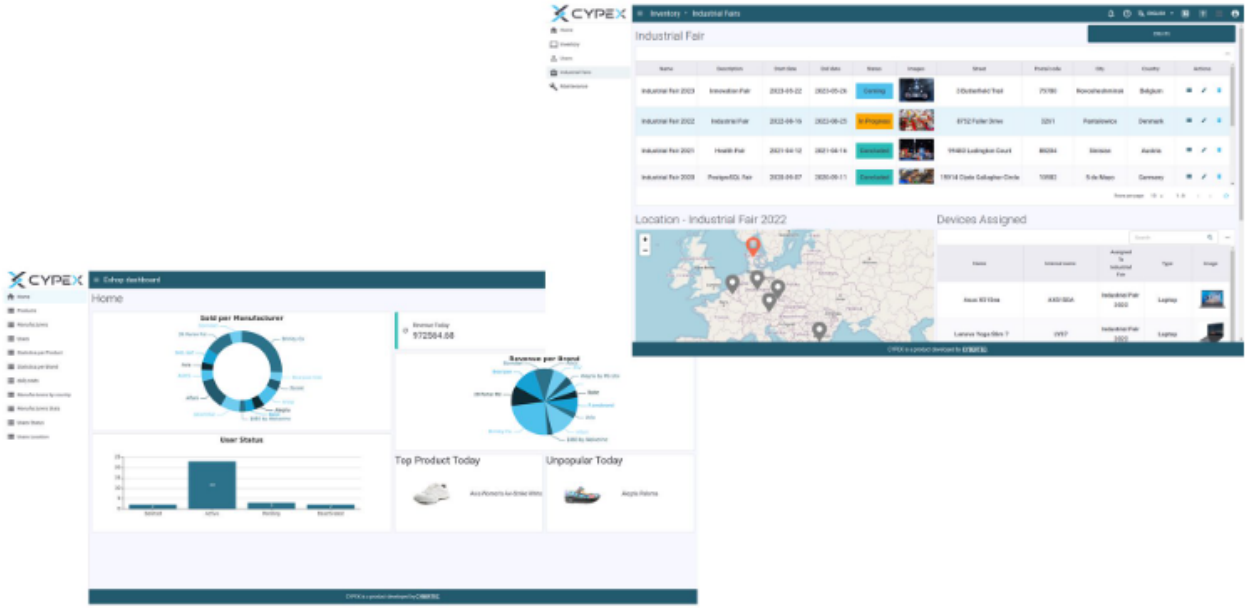
CYBERTEC
DATA SCIENCE & POSTGRESQL

CYPEX Online Academy

Learn how to use CYPEX the easy way - watch our tutorial videos. Start with an overview of the dashboard and once you understand the basics, graduate to the more nuanced aspects - for example, how to use location data, or how to insert JavaScript notation for the most flexible and beautiful charts. Not only will you come away with fresh knowledge of how to quickly build applications, you'll see how easy it is to use CYPEX for truly astonishing and instantly updatable results. Check out our video library here.



CYBERTEC
DATA SCIENCE & POSTGRESQL

Table of contents

1. CYPEX Installation and Initial Login

In this chapter, you'll be introduced to the very first steps on your CYPEX journey. We provide you with a Git repository which allows you to download, install and configure it. All containers are provided for direct use. Since CYPEX can be connected with existing databases, you'll need to provide information to tell it where PostgreSQL can be found (such as: host, port, and database name). You'll also get to know the admin panel. The admin panel is a core component of CYPEX - it allows you to gain an overview of which applications exist and to define new applications by adding workflows.

Title: CYPEX Setup and Login For the First Time

Short description:

CYPEX is tightly connected to PostgreSQL. We provide you with a Git repository which allows you to download, install and configure CYPEX. All containers are provided for direct use.

Long description:

CYBERTEC ships CYPEX in two ways. For basic needs, we ship containers that include CYPEX and PostgreSQL. However, CYPEX is often deployed to build applications for existing databases. To handle such cases, a script must be executed. The script configures CYPEX and generates credentials which can be obtained to log into the database and the GUI in a safe and secure manner.

Title: CYPEX Setup and Login Using an External Database

Short description:

CYPEX can be connected with existing databases. You'll need to provide information such as host, port, and database name to let CYPEX know where PostgreSQL can be found.

Long description:

Follow the instructions in the video to configure CYPEX to use an existing database. Keep in mind that CYPEX will deploy SQL code. The same is true for pg_timetable, which is an integral part of CYPEX and is capable of scheduling related operations.

At the end of the process, you'll have a soundly functioning CYPEX deployment.

Title: CYPEX Administration Overview

Short description:

The admin panel is a core component of CYPEX. It allows you to gain an overview of which applications exist, define new applications by adding workflows, and a lot more.

Long description:

Inside the admin panel you can control many types of behavior including permissions, LDAP authentication, application creation, auditing and user management, as well as other features. The easy-to-read dashboard shows you how many applications exist and displays some very basic information about users.

2. CYPEX Overview - create your first application in a few minutes

CYPEX applications are not built from scratch. Instead, the applications are first predicted, and then modified by the user to satisfy their needs. The application section in the admin panel helps users generate new apps. A user can select a handful of queries (or all of them) and provide some meta data. The CYPEX engine will use this information to make predictions. In this chapter, you'll learn how to create your first CYPEX application in just over 10 minutes. You'll start with a data model and end up with a full-fledged application.

Title: CYPEX - Application Section

Short description:

CYPEX applications are not built from scratch. Instead, applications are predicted and are then modified by the user to satisfy their needs. The application section in the admin panel helps users generate new apps. A user can select a handful of queries (or all of them) and provide some meta data. The CYPEX engine will use this information to make predictions.

Long description:

Before the app is actually generated, it's possible to select queries, adjust colors and handle CSS-related issues. Once the application has been predicted, it will show up in the app list. You can then export and import the app. Note that the GUI is simply a JSON document describing the layout and functionality of the application. Therefore it is easy to download and upload entire apps.

Also keep in mind that these JSON documents are stored inside the database. This implies that the data as well as the application will be part of the backup (pg_basebackup, pg_dump, etc.).

Title: CYPEX Create an application in 5 min

Short description:

Learn how to create your first CYPEX application in just over 5 minutes. This video shows the entire process from start to finish. You'll start with a data model and end up with a full-fledged application.

Long description:

This sample application gives you a guided tour through a standard CYPEX application. You'll be guided through every important stage of app creation. This includes default rendering, adding new pages and adjusting elements on those new pages.

Title: CYPEX - First Application

Short description:

Learn how to create your first CYPEX application in just over 10 minutes. This video shows the entire process from start to finish. You'll start with a data model and end up with a full-fledged application.

Long description:

This sample application gives you a guided tour through a standard CYPEX application. You'll be guided through every important stage of app creation. This includes default rendering, adding new pages and then adjusting elements on them.

3. CYPEX Application - “Database” section

Once you’ve become familiar with the admin panel, move on to the CYPEX model builder . The CYPEX model builder allows you to inspect the current data model as it is in PostgreSQL and define basic CYPEX objects such as queries, workflows, etc.

Furthermore, default queries are a core concept of the entire CYPEX system. Learn how to mark which tables are relevant to your needs with the help of a default query. The default query is assigned permissions. Depending on the extent of the user’s permissions, the query will be made available as a data source in the GUI, or will be made unavailable to that user.

Another tool we provide helps construct queries. Plain tables are often not a good data source. You’ll need joins, aggregations, windowing functions, etc. To make sure that such operations are permitted, make use of our query builder.

CYPEX offers a simple way to create workflows, an important cornerstone of the modern application-building process. Building workflows is easy in CYPEX; they allow you to move quickly from a data model to a complete application. Workflows can be changed in CYPEX without having to re-generate the application all over again. Simply go to the model builder and adjust the workflow as needed. All changes will be directly reflected in the GUI.

In CYPEX the state machine allows you to transition from one state to another. Connecting all states by hand can be cumbersome; CYPEX features the ability to “fully connect” states with each other. The idea is to have a workflow that allows for state changes from every state to every other state, ridding you of that extra work.

In order to automate various operations, triggers in PostgreSQL are offered to improve your data model.

Default lookups help to fix relational data models that would usually show ID’s as opposed to names.

One of the core features of CYPEX is adding audits to your application. Enable the “audit” feature for a relation in order to track it. The CYPEX admin panel allows you to inspect your tables as well as the relations between them. If you click on a relation, you'll also be able to see table details.

Title: CYPEX - Database Section (Model Builder)

Short description:

The CYPEX model builder allows you to inspect the current data model as it is in PostgreSQL and define basic CYPEX objects such as queries, workflows, etc. It's also possible to define permissions, audit tables and inspect data.

Long description:

The CYPEX model builder offers a comprehensive solution to manage your data models, workflows and permissions using a single easy-to-use tool. You'll be guided through all the basic steps to get started: learn how to make your first application quickly and efficiently.

Title: CYPEX Generate default query

Short description:

Default queries are a core concept of the entire CYPEX system. Often not all parts of an ER model are used by an application. To mark tables as relevant, you need to create a default query. This default query is assigned permissions. Depending on the user's permissions, the query will be available as a data source in the GUI or will be unavailable.

Long description:

Behind the scenes, CYPEX will generate a view and assign permissions. In case a table has a default view this part of your ER model will be available in the CYPEX API. A default view is therefore not just used to assign permissions and make the desired data set available as a data source in your GUI - it is also vital to make data readily available in the API so that external tools can hook up to CYPEX.

A default view is nothing more than a 1:1 representation of the underlying data. It has a name as well as a description. In the admin panel relations carrying a default view

will be marked with an icon. The default view is also visible on the right side of the screen.

Title: CYPEX Generate query

Short description:

Plain tables are often not a good data source. You'll need joins, aggregations, windowing functions, etc. To make sure that such operations are permitted, make use of the query builder.

Long description:

The query builder allows you to define custom queries. It provides a variety of features including auto completion and automatic query validation against the real PostgreSQL backend. In the editor you'll also be able to preview the data. Note that only the first handful of rows are displayed, for efficiency reasons.

After the query has been completed, you can assign permissions to it. CYPEX will assign them directly in the database to guarantee consistency between the API and the GUI.
Make sure that permissions are assigned correctly - otherwise the data source will not be available in the graphical editor. Also note that the query will be available instantly - no incremental rendering is needed to make it show up in the API and the visual editor.

3.1 Workflow

Workflows allow you to control the way data can be modified. In the visual editor, you can define states as well as state changes. Permissions can be set at every level. The entire process can be adjusted to your needs.

Title: CYPEX: Workflow overview

Short description:

CYPEX offers a simple way to create workflows, an important cornerstone of the modern application-building process. Building workflows is easy in CYPEX. Let's see how to move from a data model to a complete application.

Long description:

Workflows fuel the default rendering process by providing vital information to the application prediction engine. To create a workflow, first go to the model builder and select a table. Clicking on the three dots in the right upper corner of the relation will reveal the “workflow” entry.

A CYPEX workflow is basically a graph which consists of “states” as well as “state changes”. A workflow is always defined using a “state column”. The way it works behind the scenes is that PostgreSQL enforces a CHECK-constraint, so that the column in use can only contain values which are actually allowed. State changes will then control who is allowed to change the content of a data set, and in which way. State changes can carry permissions. What does that mean in real life? Let's take a look at a real example and see how to use a workflow to model “users”.

Users can be in one of the following states: “Pending”, “Active”, “Deactivated”, “Deleted”. Moving a user from “Pending” to “Active” is a state change (in this case it is called “Accept”). While states are just entries in tables, state changes usually turn to dropdowns or buttons in the GUI. On the PostgreSQL level, state changes are controlled by triggers which only allow certain UPDATE statements. This ensures that the database, the API as well as the GUI are all fully consistent with one another.

States as well as state changes carry titles and descriptions. The same texts are also used by the default renderer to predict an application.

Once the default application has been generated, you can see that, depending on the state a row is in, you'll see different options. Not every drop-down will carry the same options. You can't simply turn a deleted user into a pending user - the workflow does not allow that - and the GUI knows that.

Title: CYPEX Change a workflow and see the update in the app

Short description:

In CYPEX, workflows can be easily changed - without having to re-generate the application all over again. Simply go to the model builder and adjust the workflow as needed. All changes will be directly reflected in the GUI.

Long description:

We allow users to quickly change workflows. The reason is to keep the time needed to adapt to change as short as possible. Round trip times matter, therefore all changes happen in real time. It is important to note that changes to workflows are NOT related to release management inside the graphical editor. What is reverted in case an older release is brought back is its visual representation - **the data model is always available in a single incarnation**. It is technically impossible to handle various versions of a data model in a safe, consistent way. Workflows are part of the data model and therefore have to conform to the laws of the ER model rather than to the GUI. The same is true for the API which is always a reflection of the workflow. Keep in mind that workflows have been implemented on the lowest level possible to ensure that data integrity remains the highest priority.

Title: CYPEX Creating a new workflow

Short description:

In CYPEX, the state machine allows you to transition from one state to the other. Connecting all states by hand can be cumbersome. CYPEX offers the feature to “fully connect” states with each other. The idea is to have a workflow that allows for state changes from every state to every other state, ridding you of that extra work.

Long description:

In the most trivial of all cases, states are “fully connected”, which means that every state can transition to every other state. Often it is easier to start with a fully connected workflow and then modify it. If your workflow is relatively simple, this is usually the quickest approach to develop your state graph.

Of course, if you have dozens of states, fully connected workflows are not the preferred way to go. In that case, building a graph from scratch may be easier.

Title: CYPEX Change an existing workflow

Short description:

This video shows how a CYPEX workflow can be quickly adjusted via the graphical user interface, part of the CYPEX model builder.

Long description:

Simply use drag-and-drop inside your workflow editor to build a graph, to add desired states and to establish state changes. Note that the number of states defined for an object is virtually unlimited. You can add as many states as you wish. There are no usage restrictions imposed by CYPEX.

Title: CYPEX Trigger function integration

Short description:

Triggers in PostgreSQL offer vital functionality to improve your data model and to automate various operations. At this point there is no visual editor to handle triggers (yet). Triggers must be created in a database tool of your choice.

Long description:

Triggers in PostgreSQL can be defined on tables as well as on views. While triggers on tables are far more common, triggers on views are equally important. Why does it matter? In CYPEX, queries are essentially views on tables. Therefore it's necessary to create triggers to handle updates on views which are not auto-updatable.

Notifications are also an area which require triggers. Sometimes you may want to notify somebody about a state change in CYPEX (e.g. a contract is signed, etc.). Triggers are an ideal way to handle those cases.

In PostgreSQL a trigger is always based on a function, which means that the same trigger can be used to serve multiple triggers at once. That allows for a great deal of code abstraction not available in other database engines.

Also keep in mind that in PostgreSQL, triggers are fired in alphabetical order - which is important if you have more than one trigger defined on the same table. It is also noteworthy that a standard row-level trigger has to return NEW or OLD, depending on the operation. Those are predefined variables which carry the row the trigger operates on. INSERT has to return NEW or NULL. If NULL is returned, the operation is ignored. DELETE has to return OLD (= row to be deleted) or NULL (in case nothing is supposed to happen). In case of UPDATE, NEW is usually returned, which represents the row as it is after the desired changes.

Title: CYPEX Database section default lookup

Short description:

The concept of “normalization” is key to understanding SQL. The problem is that in many tables, “names” are not stored as such but are represented as ID’s pointing to some other table. This leads to a usability problem because in a graphical interface, you want to see names rather than ID’s. Default lookups help to fix this problem.

Long description:

Displaying IDs instead of human readable names is far from ideal. In a real application users want to see texts instead of ID's. However, in a large data model it can be pretty cumbersome to manually change all those things in the end user GUI, write joins or come up with some other tricks to work around this inherent relational idea.

The solution in the CYPEX world is the concept of “default lookups”. For each table, it’s possible to define a field which should be displayed instead of the ID. The default renderer will inspect your model and automatically put names wherever it finds ID's that do have a default lookup. During the resolving process your foreign key relation will be inspected. In general the ID is resolved by the GUI to ensure that no UPDATE triggers on joins are needed. In case the name has to be processed in a more sophisticated way by the GUI, the use of “custom expressions” is recommended.

Title: CYPEX Database section auditing

Short description:

“Security is king” - this is especially true if you are dealing with critical data. One of the core features of CYPEX is the ability to add auditing to your application. Simply enable the “audit” feature for a relation to audit it.

Long description:

In case auditing is enabled for a relation, CYPEX will deploy a changelog trigger on the desired table. It will track INSERT, UPDATE, DELETE as well as TRUNCATE. All changes found by the mechanism will end up in a CYPEX system table for later inspection. Data will be stored in JSON format to ensure a uniform storage method and to make searching the audit tables easier.

Check out the audit trail in the admin panel. The audit section will reveal the history and allow for efficient searches.

Title: CYPEX Database section table details

Short description:

The CYPEX admin panel allows you to inspect your tables as well as the relations between them. If you click on a relation you'll also be able to see table details.

Long description:

The table details section of your relation will reveal the list of columns. However, you'll also be able to take a look at the data itself. The CYPEX admin panel displays the first 100 rows in your table to give you a brief insight into which data is available in your database. It is only meant to be a short overview.

4. CYPEX Application - “Authentication” section

The CYPEX model builder allows you to configure the authentication strategy of your application and helps you to configure your security model.

Creating new users in CYPEX is easy. Simply go to the “Users” section in the admin panel. Add and remove users in the "Users" section within the admin panel. There you can decide who is allowed to make use of your applications. Create and add roles as you need it in the admin panel.

The application authentication section allows you to customize your login screen. You can set the logo as well as the title.

In a large corporation, Single-Sign-On is the authentication method of choice. CYPEX supports SSO authentication and allows you to connect user management to LDAP.

By including ready-made extensions we’ve drastically reduced the overhead of high-usage services (e.g. addresses and country lists). CYPEX has all the means to do more with less time.

And last but not least, there are 2 types of repositories: extensions and ready-made apps. The structure of these two types of repositories are slightly different. Depending on the type of repo, it will either show up in “repository applications” or in “extensions”.

Title: CYPEX Authentication section overview

Short description:

The CYPEX model builder allows you to configure the authentication strategy of your application and helps you to configure your security model in great detail.

Long description:

The user list will show which logins are possible and which ones are not. You'll be able to add users to the scenario and configure the authentication method. What is important here is the ability to use LDAP as backend technology to handle authentication. Note that LDAP is ONLY used for authentication - in case new users are added to LDAP, CYPEX does not know and does not care. A user has to exist in CYPEX to function properly.

Title: CYPEX Managing internal user accounts

Short description:

Creating new users in CYPEX is easy. Simply go to the “Users” section in the admin panel. There you can add and remove users via the graphical interface. Decide who is allowed to make use of your applications.

Long description:

To log into CYPEX, the user must be valid and set to active. Users can be created in the admin panel. Please note, a user can be identified using an email address or with the “username”. Both strings can be used to log in to the system. The username is optional.

Usernames and email addresses are mapped to database users. Many email addresses / usernames can be mapped to the same database role. The goal here is to use PostgreSQL side roles as grouped categories to assign the same permissions to various CYPEX frontend users. In most cases, this comes in handy. In a real-world scenario, 100 bookkeepers identified by email might be mapped to the same server-side user.

Title: CYPEX Manage roles

Short description:

It's easy to add roles in the admin panel. Simply go to "Authentication -> Roles" and create whatever role is needed.

Long description:

Note that while it is possible to create roles, this feature will be expanded in the future to make it more powerful: it will allow for the definition of more parameters and settings. For now, simply follow the instructions to create a role.

Title: CYPEX Login settings

Short description:

The login screen usually shows the CYPEX logo. However, the default login screen can be customized. You can set the logo as well as the title.

Long description:

Modifying the standard login screen can be done by using the "Login Settings" in the "Authentication" section in the admin panel. Currently, two parameters can be changed: The logo and the title of the page. While changing the text is easy, changing the logo requires a bit more explanation: The logo you are using has to be in the public folder of the web server. Otherwise, CYPEX won't find the image.

Title: CYPEX Login using LDAP server

Short description:

In a large corporation, Single-Sign-On is the authentication method of choice. CYPEX supports this type of authentication and allows you to connect your user management to LDAP. To facilitate the setup, you need to take care of various issues.

Long description:

Configuring LDAP in the graphical user interface (admin panel) is not sufficient. You'll need to enable some settings in the editor as well. First you need to enable LDAP in the CYPEX container. Do that by opening the ".env" file and set LDAP=on. Then update the container by running make up-d.

Once this is done, proceed with the graphical interface and insert the LDAP information. LDAP will serve as a full replacement for other authentication methods.

Title: CYPEX Repository configuration

Short description:

Typically, ER models are built by hand. However, this is sometimes not very efficient. Why create tables to handle (e.g.) addresses time and again? Why define country lists over and over? By including ready-made extensions we've drastically reduced this overhead. CYPEX has all the means to do more with less time.

Long description:

Git repositories allow users to add extensions to CYPEX data models. Configuring the repositories will allow CYPEX to clone them and present the list of available extensions graphically, so that deployments are made easy.

Prior to saving, it is highly recommended to test the connection - which is possible using the button on the screen.

Title: CYPEX Explore repository content and install applications

Short description:

There are 2 types of repositories: Simple extensions and ready-made apps. The structure of those two types of repositories are slightly different. Depending on the type of repo, it will either show up in the “repository applications” or in “extensions”.

Long description:

Application repositories are identified by a file called “description.json”. It turns a repository into an application repository. Here is a sample that shows what this file might look like:

```
{
    "name": "cypex_demos",
    "shortName": "CDEM",
    "description": "Cypex application distribution demo repository",
    "fileVersion": "0.1",
    "applications": [

  ]
}
```

This is a full example with an application located in "sourcePath": **demo_eshop/release_1**
This application is organized in three files **:**

"fileNames": ["**001_base.sql**", "**002_sample_data.sql**", "**003_model.sql**"]

```
{
    "name": "cypex_demos",
    "shortName": "CDEM",
    "description": "Cypex application distribution demo repository",
    "fileVersion": "0.1",
    "applications": [
```

```
  {
   "name": "demo_eshop",
   "description": "Cypex demo eshop application",
   "creator": "Hans",
   "owner": "Cybertec",
   "firstReleaseDate": "2019-07-26 11:33:56",
   "text": "Cypex demo eshop application",
   "releases": [
    {
     "sourcePath": "demo_eshop/release_1",
     "version": "0.1",
     "fileNames": ["001_base.sql", "002_sample_data.sql", "003_model.sql"],
     "text": "Cypex demo eshop application",
     "date": "2019-07-26 11:33:56",
     "uninstallScript": "demo_eshop_uninstall.sql"
    }
   ]
  }
]
}
```

5. CYPEX Application -"File Management" section

File storage that can handle user-uploaded files is a need in every web-editor. You can quickly integrate photos into all upcoming apps while keeping the flexibility to backup data and programs as one by fully utilizing the CYPEX file store.

Title: CYPEX File management section

Short description:

Two ways to handle files are discussed in this video: First of all, you can upload files in the admin panel and use those files across the application. The second method is to upload those files directly inside the application designer (= GUI editor).

Long description:

CYPEX offers a file store which is capable of managing files uploaded by the user. By making full use of the CYPEX file store, we're easily able to integrate images into all future apps while maintaining the ability to backup data and applications in one go.

6. CYPEX Application -"Audit" section

Critical applications need auditing. CYPEX has onboard means to handle this, out-of-the-box. Click on relations in the ER editor and enable the audit module. CYPEX will configure all that is needed for you.

Title: CYPEX Audit section

Short description:

Critical applications need auditing. CYPEX has onboard means to handle this, out-of-the-box. Click on relations in the ER editor and enable the audit module. CYPEX will configure all that is needed for you.

Long description:

You can audit tables by deploying generic change log triggers in CYPEX, which sends a copy of your changes to a log table. The visual editor then allows you to track all changes in a table, and see visually what has changed where (diff format).

It's noteworthy that within CYPEX history, table data is stored in JSON format to ensure easy analysis and to guarantee that data is stored in a uniform, easy-to-process format which is not prone to break, in case data structures change.

7. CYPEX Application - “Data API” section

CYPEX automatically exposes your data as an easy-to-use API which allows you to integrate your database with external platforms in a straightforward manner. The APIs automatically keep up to date in case changes are made to the data model.

Title: CYPEX Data API

Short description:

CYPEX automatically exposes your data as an easy-to-use API which allows you to integrate your database with external platforms in a straightforward manner. The APIs automatically keep up to date in case changes are made to the data model.

Long description:

The admin panel offers a page which helps you to test your API. Note this is not a test environment - all changes are going to happen live in the database. Therefore caution is recommended when using the visual interface or the API directly. Mind that to use the API, you need to authenticate using the same mechanism as the CYPEX GUI’s (we use JWT).

8. CYPEX Application - “Add-ons” section

CYPEX allows you to hook up to a repository and load ready-made SQL fragments directly into your application, allowing you both to perform basic common tasks more quickly and to achieve a higher level of quality at those tasks.

Repository applications are applications which can be directly downloaded into your CYPEX deployment. This video tutorial shows how such applications can be made.

Title: CYPEX Add-Ons

Short description:

CYPEX allows you to hook up to a repository and load ready-made SQL fragments directly into your application, allowing you both to perform basic common tasks more quickly and to achieve a higher level of quality at those tasks.

Long description:

CYPEX add-ons are PostgreSQL extensions which contain ready-to-use fragments such as addresses, currency lists, models to store product data, unit conversion and much more. Adding extensions makes data models more consistent, and by using them the time it takes to produce an application can be reduced significantly. CYPEX supports full version control as well as access to standard DevOps-related pipelines.

Title: CYPEX Create a repository application

Short description:

Repository applications are applications which can be directly downloaded into your CYPEX deployment. This video tutorial shows how such applications can be made.

Long description:

First, create a directory for your new application. In this directory, you need an SQL file which contains the data model of your application. Keep in mind that the CYPEX role “authenticator” must have access to your tables, otherwise CYPEX has no permissions to manage those relations directly. The next thing you need is an “uninstall” script. The idea here is simple: When you deploy an application, you also have to figure out how to get rid of the application after it has fulfilled its purpose. The important part is that running the install and the uninstall script one after the other should leave you with an empty database.

Ideally you should split the model and the sample data into separate files. This is usually more convenient as it is easier to remove the test data later or to manage your repository in general.
Once this is done you can create a file called description.json. It will contain a lot of metadata CYPEX will need to manage your application nicely. The easiest way is to copy your sample files, and take it from there.

Finally, you can add your application to Git and deploy it easily using the graphical user interface provided by CYPEX.

9. CYPEX Application “Application Designer” section

In this chapter, you'll learn how to create a sample application based on an existing database model.

Watch our videos and have a look at the huge variety of field types (e.g. boolean with "true" & "false", code input fields, date & time inputs, number fields, slide input, text elements, multiple file input fields, array text fields, buttons, subform tables, ...). Many of these field types are working with autocomplete and syntax highlighting.

Furthermore, CYPEX offers special design and grouping functions. A container element allows you to group elements wihin a grid. They can be conditional as well. Markdown fields are a generic way to add content to the screen. They can be linked to other elements on the page and are therefore an easy way to display things and to easily generate content on the fly. Not to forget about images! They are one of the most basic design elements on any page. They are not only used statically but are often subject to dynamic behavior. CYPEX lets you work with color fields. CYPEX is able to display a square containing a color to indicate information on the page.

On top of that, CYPEX lets you edit JSON documents using a special GUI element. The key advantage is that the GUI element assures that the JSON document sent to the backend is always valid which helps to avoid errors.

CYPEX offers many more timesaving functions, find out about:

- Internal and external link fields in order to navigate inside a CYPEX application or call a link outside of your application.
- Intelligent clipboard
- Modal dialogs are elements which display other elements inside the dialog box (mini pages on top of your main page)
- Tabs - use the size of your screen more effectively by grouping various elements in a simple, compact form. Manage this entire group of elements at once.
- Many different chart-visualisation options
- Circular progress elements
- Usage of Geo data (= GIS)

Title: CYPEX Application Overview

Short description:

In this video, you'll learn how to create a sample application based on an existing database model. Learn how to use the default renderer and understand how to adapt the design visually.

Long description:

You'll be guided through the standard rendering process. You'll see how to select from a list of queries and turn the data model into an (almost) ready-to-use application using just a handful of clicks. Also note that CYPEX will try to process your query names in a smart way to reduce the number of post-processing steps to as few as technically possible.

State: draft done https://cybertec.atlassian.net/browse/CN-851

Title: CYPEX Data Display element

Short description:

A data display element is a visual element which allows you to prominently display a piece of information on the screen.

Long description:

Often it is necessary to emphasize a certain piece of information on a page. A data display element calls attention to specified information. It allows you to quickly and prominently display a chunk of information - including a logo - on the screen.

Title: CYPEX Array Text Input

Short description:

Text array fields allow you to display various small pieces of information in one element. In this video you'll learn how to use text array fields in a more advanced way by combining text array fields with other GUI elements.

Long description:

Combining elements on the screen to work together is one of the key tasks when building a visually appealing application. This video shows how to use text array fields in a typical and practical context. Use a table along with text array fields to associate groups with a row of data. Note that this type of use case is the most common one in conjunction with text array fields.

Title: CYPEX Autocomplete Input Element

Short description:

Autocomplete fields are a way to quickly gather information from the end user by suggesting values already. This type of GUI element has to be combined with other elements on the page to unleash their full power.

Long description:

In CYPEX autocomplete fields help the user find and produce information more quickly. Behind the scenes, the field is configured to ensure that it knows how to retrieve information from the web backend. The video shows how to configure a filter in a simple way to tell CYPEX how to fetch the data

Title: CYPEX Boolean Input Element

Short description:

Boolean fields are some of the most basic input fields. They display only “true” or “false”. However, they are important in conjunction with other GUI elements.

Long description:

Boolean field elements can achieve two things: On the one hand you can display boolean values and make them more visually appealing (a field rather than “true / false” as string). On the other hand, they serve as switches to turn things on and off. In the second role, boolean inputs are often used as connected objects. With the help of expressions, you can control the behavior of an entire page.

Title: CYPEX Code Input

Short description:

CYPEX provides a special type of input field which is able to handle code. We support a variety of languages and enable syntax highlighting.

Long description:

Code fields are often needed to give users a more convenient way to type code. In particular, syntax highlighting is a key feature of our editor. We support a variety of programming languages and make it easy to enter such data in a visually appealing way.

9.1 CYPEX Application Layout Elements

Layout elements give the application its look and feel. They can be placed on the playground inside a grid and data elements can be placed inside these elements.

Title: CYPEX Container

Short description:

A container element allows you to group elements. Basically, a container element contains other elements. This allows you to easily turn on / off complete subsections of your page. Just like other elements, a container sits on the grid.

Long description:

By using containers, you can turn GUI elements which are usually independent from each other into larger groups which can all be controlled at once. Examples of what you might find inside a container are tabs or complete forms.

Title: CYPEX Conditional Container

Short description:

A conditional container is a container that is hidden in case a custom expression does not return true but false. The way it is used is to access the content of other elements and dynamically decide how to respond.

Long description:

If you want to decide whether a conditional container is shown or hidden, you need to make use of custom expressions - which are a core component of CYPEX. The way it works is that inside your graphical config element, you can feed a JavaScript snippet into the configuration. This expression will evaluate and in case it returns true, the conditional container will be shown. Otherwise it will be hidden from view.

The custom expression can access any element on the page, even including hidden elements.

Title: CYPEX Markdown Text Element

Short description:

Markdown fields are a generic way to add content to the screen. They can be linked to other elements on the page. They make it easy to display data and to generate content on the fly.

Long description:

Markdown fields serve many purposes. The default use case is to statically display data on the screen. However, there is a lot more to Markdown fields that meet the eye. Often you can use Markdown fields to display generic content. By connecting the markdown field to a table (for example), you can display dynamic content and change the behavior of any field which depends on other elements.

Title: CYPEX Image Element

Short description:

Images are one of the most basic design elements on any page. They are not only used statically, but are often subject to dynamic behavior.

Long description:

Just like Markdown fields, CYPEX images can serve two roles: They can either display static, constant content or they can link to some data source and display content dynamically, depending on your needs. Again, use custom expressions to make dynamic content happen.

9.2 “Form” & “Display” - List Elements

Everything to help manage data inputs many fields such as: text input, date-time, drop-down, multiline text and more.

Title: CYPEX Color Field

Short description:

CYPEX features color fields. You can display a square containing a color in order to indicate information on the page.

Long description:

Color fields can be used in conjunction with other elements to conditionally display colored squares on the screen. In CYPEX this is often done to display different kinds of status information.

Title: CYPEX Date-Time Input

Short description:

Date and time inputs are used to display and modify time-related data. You can use them in various ways, from the simple to the more complex.

Long description:

Date and time are essential and are part of basically every practical application containing forms out there. In CYPEX you use those fields in various, flexible ways to either gather user information or simply present data to the end-user.

Title: CYPEX JSON Input

Short description:

JSON documents can be edited in CYPEX using a special GUI element. The key advantage is that the GUI element assures that the JSON document sent to the backend is always valid which helps to avoid errors of all kinds.

Long description:

PostgreSQL provides significant support to handle JSON documents. However, it's important to avoid sending broken data to the backend by allowing untrained users to access special input fields capable of handling JSON documents. The GUI element will produce valid JSON and thus reduce the number of errors coming from the database backend.

Title: CYPEX Markdown Input Element

Short description:

A Markdown field allows users to quickly display data in a formatted way. The main benefit of Markdown is that it makes it easy to apply simple text formatting and to display information on the screen.

Long description:

In this example, you'll learn to connect a Markdown field with other elements on the screen to produce interactive applications. The main advantage of Markdown lies in its simplicity. As you can see in this video, you can easily use Markdown to change the layout of a row inside a table by applying it to format text.

Title: CYPEX Number Input Element

Short description:

Number fields are one of the most powerful design elements in CYPEX. This type of element is not just capable of displaying numbers but supports all kinds of magic on how those numbers are displayed.

Long description:

Numbers can be displayed in various formats and the numeric elements in CYPEX support most of this magic out-of-the-box. Enrich the way data is displayed by assigning advanced format instructions to numeric fields to achieve the best results possible. Link elements to fetch data from other parts of the page. In the example in the video, you can even see how the resolution of images can be displayed on-screen within seconds.

Title: CYPEX Slide Input

Short description:

Sliders offer an easy way to show numeric input or to gather intervals from the users. Sliders can be configured in various ways to handle log scale as well as other data.

Long description:

Sliders are especially useful if you want to make the user input data which is only supposed to be within a certain range. In CYPEX sliders are a really powerful way to handle those cases. There are various configuration options to make sliders more beautiful and useful over all.

In this video you'll learn to get the best out of sliders in the most simplistic way possible.

Title: CYPEX Subform Table

Short description:

Subform tables are needed to edit 1:n relationships conveniently. They help you to avoid switching around between pages - which greatly improves the efficiency and usability of your application.

Long description:

In case of 1:n relationships, you may want to display the “1” as well as the “n” side on the same screen. This video shows you how to create such forms and how to connect objects with each other to interact nicely.

Title: CYPEX Text Input Element

Short description:

Text elements are one of the most common building blocks in CYPEX. They’re heavily used by the default rendering engine to display data in the most basic form.

Long description:

In CYPEX, text elements are often the backbone infrastructure used inside forms and containers. Most kinds of data can be interpreted as text (apart from images and a few others) and therefore basic text elements can serve as the most basic method to display data.

Making the layout fancier, however, is of course possible by using custom expressions which can dynamically determine the content of fields depending on other elements on the page.

Title: CYPEX File Input

Short description:

File inputs allow users to upload files into the CYPEX file storage system. You can control permissions as well make CYPEX aware of the type of file you want to be handled.

Long description:

This video shows how to make full use of the file management infrastructure in CYPEX. You'll learn how to limit the inputs allowed and you'll be guided through other aspects such as permissions. Finally, use the file upload in combination with other elements to demonstrate the interaction of GUI elements.

Title: CYPEX Multiple File Input

Short description:

Uploading more than one file at a time is done using the “multi file input” element. In contrast to a normal file upload, here you can process many files at once.

Long description:

Similarly to when uploading a single file, with multiple file input you can define the MIME type to control what can be uploaded and how. Multi-file input elements also support permissions and are able to handle various other features which are identical to what single file uploads can do.

Title: CYPEX Action Buttons

Short description:

Buttons are one of the basic elements of every application. In CYPEX, action buttons offer a rich set of functionalities and allow you to perform all kinds of pre-defined and not-predefined actions.

Long description:

A CYPEX action button gives you the ability to perform all kinds of operations on your page. There are various types of actions including “Refresh page”, “Show notification only”, and “Navigate”, among others. Depending on what you want to do, many different behaviors can be configured.

Title: CYPEX Array Text Field

Short description:

Array text fields are elements which can be used to display a set of data as small little elements inside a single field. They are often used to show, for example, group memberships and lists of categories.

Long description:

Using text array fields enables you to display various pieces of text as a single element. This comes in handy if you want to display group membership or if you want to list attributes associated with an element. Often this is also used to display a lift of emojis or other small fragments of data.

Title: CYPEX Boolean Field

Short description:

In PostgreSQL boolean fields will return “t”, “f” or NULL. None of these representations are user-friendly. The way to display boolean data in CYPEX is to use dedicated boolean fields.

Long description:

Boolean fields can hold 3 different values: true (= checkbox is ticked), false (= checkbox is not ticked) as well as NULL (= checkbox is crossed out). We can either display data directly or control the content of the boolean field based on a custom expression as used by basically all other CYPEX elements.

Title: CYPEX Call Button In Action

Short description:

Call buttons are a GUI element which allow you to call a server-side function. Typically, these elements are used to trigger server-side business logic in a user-friendly way.

Long description:

A function exposed in the cypex_generated schema will be available in the API, allowing you to hide server-side functionality behind a button. This video will show you how to write a server-side function and how to use it to provide the business logic behind a button.

Title: CYPEX Circular Progress

Short description:

Often you may want to display some kind of progress information. Circular progress elements serve exactly this purpose. Use it to display a number in a circle and show progress.

Long description:

A circular progress element can display a value between 0 and 100 to indicate progress as a percent value. The element can be fed by a custom expression. Progress is displayed visually in a circle.

Title: CYPEX “Copy to Clipboard” Buttons

Short description:

Often it is necessary to copy elements to the clipboard. Manually marking elements using a mouse can be cumbersome and time-consuming. Therefore having a method to copy other elements can allow you to perform tasks more quickly.

Long description:

Using a “copy to clipboard” button allows you to copy the content of some fields (or a modification of such content) to the clipboard. It makes it easier for people to move data around.

Title: CYPEX Color Field

Short description:

Color inputs can be used to display colors or give users the ability to configure them.

Long description:

Color input fields allow users to display colors and fetch input from the users. Colors have to be fed in hex format (leading #). The size of those elements are configurable.

Title: CYPEX Date-Time Field

Short description:

Date / time fields allow CYPEX users to display time, dates as well as intervals.

Long description:

Time / date can be displayed as absolute value or as relative value. In addition to the format it is also possible to define the time zone you want to see.

Title: CYPEX External Link Field

Short description:

While internal link fields allow you to navigate within a CYPEX application, an external link field is in charge of calling a link outside of your application.

Long description:

In case it becomes necessary to embed an external link into the application CYPEX has the right GUI element for you. External link elements allow you to basically call every kind of link and leave the application if necessary.

Title: CYPEX Form

Short description:

Forms are a way to allow users to enter and edit data. The purpose of a form is to group elements in such a way that the system can correctly choose what to submit to the backend.

Long description:

A form allows you to gather various pieces of information and submit them to the server all in one go. The form is therefore a group of elements that is supposed to be saved together. In CYPEX, forms are either auto-generated during the initial stage (app prediction) or can be added later, in case you want to add input data.

Title: CYPEX Internal Link Button

Short description:

Internal link fields allow you to move around inside a CYPEX application.

Long description:

Use internal link fields to navigate inside a CYPEX application. In this video, you'll learn how to create a button which allows you to quickly navigate to the "home" section of your application. The video features a typical use-case which shows you how to use internal link fields in CYPEX.

Title: CYPEX Internal Link Field

Short description:

Normal links can be used to navigate to pages outside of the CYPEX ecosystem. However, often you'll want to navigate inside CYPEX and jump quickly from one page to another. That's the purpose of internal link fields.

Long description:

Internal link fields are used to navigate inside a CYPEX app. You can either link to a static page or be more dynamic and calculate the link on the fly to link to any page containing any content. Simply use the configuration bar of the link field to control the behavior of this important GUI element.

Title: CYPEX JSON Field

Short description:

JSON fields are a powerful way to make JSON editing easier and less error prone. In this example you'll learn to see how a JSON input field can be associated with other elements on the screen to make the application more powerful.

Long description:

JSON input fields have the ability to validate JSON before it is actually sent to the server which greatly simplifies the process of editing. It is also possible to avoid a handful of errors this way. In this video you'll learn how to connect JSON input fields with other elements on the screen to make interaction of elements even more powerful.

Title: CYPEX Modal Dialog

Short description:

Modal dialogs are elements which are used to show other elements inside the dialog. In other words: You can use them to display mini-pages on top of your main page.

Long description:

A modal dialog will open a little mini-page on top of your main page. It will again contain a grid which can be filled with other elements. The modal dialog is ideal to produce little subforms in order to quickly edit an entity. Its main advantage of is that you don’t have to change the page when new things are added to the database.

Title: CYPEX Number Field

Short description:

A number field allows you to display numeric data in a user-defined format.

Long description:

A numeric field gives you the ability to display a number in almost any format commonly needed in applications. It is a powerful tool for the presentation of numbers.

Title: CYPEX Table

Short description:

Tables are the backbone of every CYPEX application. They are able to display all kinds of data in an easy-to-read form.

Long description:
It's important to understand that an entry in a table is an "expression", which means that you can modify what is displayed using the custom expression machinery of CYPEX. There are also features which allow you to automatically refresh a table, and to easily interact with other elements.

Title: CYPEX Tabs

Short description:

Tabs are visual elements which allow you to use the size of your screen more effectively. The core idea is to give users a chance to group various elements in a simple, compact form and handle this entire group of elements at one.

Long description:

By adding a tab you can make your application way more compact and easier to use. Simply drag a tab element onto the screen and start to add elements to it. While it is possible to influence the layout of your tabs it is also possible to add a basically infinite number of elements to every subpage of your tab.

Title: CYPEX Text Fields

Short description:

Text fields allow you to display text in various formats. You can apply various style changes.

Long description:

It's possible to display text in various colors and to decide how it is arranged (centered). In addition to other capabilities such as being able to control the layout of the GUI element, you can dynamically generate the content as needed.

Title: CYPEX Text Input

Short description:

A text field allows you to gather raw text. It is one of the easiest to understand, yet powerful GUI elements.

Long description:

Text fields can do more than just gather text from the end-user. They also allow you to ensure that only certain strings are accepted. This is especially important if you want to check for email addresses or other types of input.

9.3 Charts

Visualization is an important aspect if you want to present data. In this section you'll learn about charts and how to use them.

Title: CYPEX Bar Chart Element

Short description:

Bar charts allow you to visualize time series and many different kinds of statistics in a clearly understandable fashion.

Long description:

In CYPEX, a bar chart can serve as a fundamental building block for dashboards and similar tools. Bar charts provide a quick, colorful, easy-to-understand way to display data.

Title: CYPEX Custom Chart Element

Short description:

Custom charts are one of the more powerful GUI elements. However, they require more configuration and more manual coding than other elements.

Long description:

A custom chart does not contain information and layout by default. You must configure the chart manually and write code to make things happen. Behind the scenes, we use the Apache echart library to display data. Please check out the echart documentation to find out more about configuring this type of chart.

Title: CYPEX Line Chart Element

Short description:

Line charts are one more vital GUI element which can be utilized on a dashboard or on other pages that have to display data in an intelligible manner.

Long description:

Line charts in CYPEX are highly customizable and allow you to display many different kinds of data in a variety of ways.
Behind the scenes, we use the Apache echart library to display data. Please check out the echart documentation to find out more about configuring this type of chart.

Title: CYPEX Pie Chart Element

Short description:

Pie charts are widely used GUI elements which are often seen in dashboards or other overviews. They are easy to create and can be customized heavily. Often pie charts use data coming from a query directly to display information quickly.

Long description:

In CYPEX and most other applications pie charts are the workhorse to display aggregate data in a user friendly way. Select a data source and adjust the layout of the chart as you like. The visual editor allows you to change the colors of your chart, use text descriptions and a lot more.

9.4. 3rd Party

In addition to the GUI elements you have just seen, there are many more widgets which are capable of displaying data. In this section you'll learn about GIS-related elements.

Title: Leaflet Map GeoJSON Field

Short description:

The purpose of a Leaflet Map is to display data represented as GeoJSON on a map.

Long description:

Leaflet Maps are a powerful tool to visualize geometric data in a simple way. This type of elements is configurable and can be customized to your needs. The important part is that data has to be returned by the backend as GeoJSON. Other formats are not accepted. Use PostGIS functionality to create GeoJSON.

Title: CYPEX Geographic Inputs (Managing GIS Data)

Short description:

Geo data (= GIS) data is becoming more and more relevant. That’s why CYPEX offers powerful tooling to gather GIS data using a graphical user interface.

Long description:

This video explains how to create a GIS application from scratch, using maps in CYPEX . You'll see everything from the creation of the data model all the way to the final application. Note that CYPEX supports PostGIS out of the box and allows you to display GeoJSON documents with relative ease.

10. CYPEX Application - Design Best Practice Section

Have a look at our showcases:

- How to add autocomplete fields to an application
- How to show current user data
- How to send emails using pg_timetable
- How to link objects
- How to refresh after a form has been saved

Title: CYPEX Adding Autocomplete field to an application

Short description:

Dropdown fields are often used to allow people to select a value from a small set of data.

Long description:

This video explains how a dropdown field can be added to an existing application by linking GUI elements together. Learn how to add such a field and to connect the dots. To feed a dropdown element with data, you can use standard data sources (queries).

Title: CYPEX Show Current User Data

Short description:

This sample app shows an example that demonstrates how internal user data can be displayed.

Long description:

CYPEX does a great deal of internal user and authentication management. This sample will show how to display this data and make information accessible to end-users quickly. You'll learn how to build an application rapidly using the admin panel to create queries and the visual editor to layout the application.

Title: CYPEX Sending Email with PG_Timetable (Sample app)

Short description:

pg_timetable is a powerful scheduler for PostgreSQL. It's part of CYPEX. It can perform many tasks including running complex SQL tasks and sending emails. In this video, you'll learn to use pg_timetable to send emails from CYPEX.

Long description:

In pg_timetable, sending an email is a so-called "builtin" task. As soon as pg_timetable has all the configuration to send an email, use SQL to populate the pg_timetable infrastructure to send email. Then build a CYPEX form to display all the information needed to gather the information to send emails.

Title: CYPEX Creating Related Table Views (Sample App)

Short description:

Linking objects together is a core concept of the application builder. In this video, you'll learn how to link objects.

Long description:

This video shows how two tables can be linked together and made dependent on each other. The basic idea behind linked objects can be applied to other types of GUI elements as well. The method is the same, which gives a lot of flexibility to app developers making full use of the graphical editor.

Title: CYPEX Refreshing After Saving a Form (Sample App)

Short description:

In case changes are made to an object on the screen, it's necessary to define how other elements react to it. This video shows how you can refresh after a form has been saved.

Long description:

The real challenge when building an application is to define the interaction between elements on the screen. CYPEX has various methods to configure the interaction of objects. This video shows the most common dependency between objects: having one object refresh in case some other object is changed.