

**Gregor de Cillia**  
**Alexander Kowarik**  
**Bernhard Meindl**  
Statistics Austria

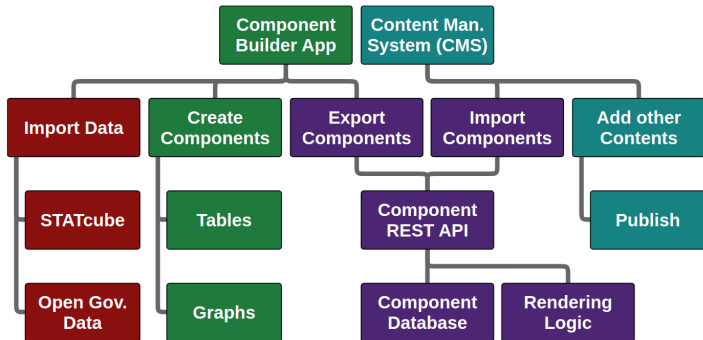
Vienna  
November 2021

# Building a Content Delivery Pipeline for a Website in R

Embedding web components from R into a content management system

**Statistics Austria** is currently working towards the release of a new website. An important part of it, namely **graphs** and **tables**, will be created using **R**

Content creators will create **graphs** and **tables** via a **shiny app** and transfer them to the content management system (**CMS**). The transfer uses a **plumber** REST API under the hood



## Parts of the presentation

1. Showcase the **app** and **components**
2. Remarks about the **data import**
3. Mechanism to **transfer** components from the **app** to the **CMS**

The components that can be delivered by the pipeline are **graphs** and **tables**

Both the graphs and tables are **interactive** in the sense that they provide hover and/or click events

- Graphs are generated with **highcharts.js** and the corresponding R package **highcharter**
- Tables are created using **datatables.js** which provides features like **sorting**, **searching** and more

Interactive demos can be found in the **html version of these slides**

The component builder **app** is a **shiny** application which is hosted on **RStudio Connect**. It guides the content creator through the component generation process

- Data can be **imported** from different sources
- Components can be **created** and **previewed**. Sensible defaults make it possible do this with a few clicks
- An **export menu** makes the component accessible for the **CMS**

A demo video of the app can be found in the **html version of these slides**

Currently, there are two datasources which are compatible the component builder **app**: **STATcube** and **Open Government Data**

Both data sources are imported using the open source package **STATcubeR** which defines a common **interface** for the data sources

```
# online docs: statistikat.github.io/STATcubeR  
remotes::install_github("statistikat/STATcubeR")
```

The **Open Data Portal** of Statistics Austria provides a series of datasets following the **open data** principles

About **300 datasets** are compatible with the component builder **app** which provides a coverage of about **95%**

OGD data can be imported from the **app** by choosing one of the datasets from a selection menu

More information can be found at [https://statistikat.github.io/STATcubeR/articles/od\\_table.html](https://statistikat.github.io/STATcubeR/articles/od_table.html)

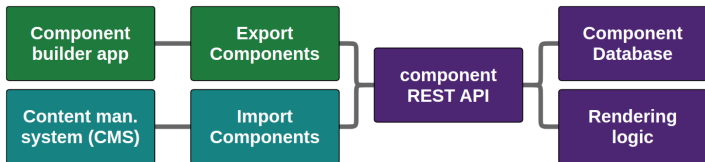
STATcube allows you to **generate** custom tables from databases and **export** them in different formats

One export option, the “**json API request**”, is compatible with the **component builder app**. The exported json file can be provided to the app via a **shiny fileInput**

The app then fetches data from the **STATcube REST API**

More information can be found at **https://statistikat.github.io/STATcubeR/articles/sc\_table.html**





In order to make the **graphs** and **tables** available for the **CMS**, a **REST API** is used. The **app exports** the components to a **database** and the **CMS imports** the components

The API is built with **plumber** and hosted on **RSCoconnect**

When a user **exports** a graph or table, the app performs a **/POST** request against the component API

The component is submitted as a **binary** object in the request body and added to a **BLOB** field in the **component database**

The database also captures **metadata** about the component such as the author and a timestamp

The `/GET` method of the **component API** lists all components in a json format. A tabular representation is provided below

```
httr::GET("rsconnect.local/component-api")
```

Key	Created.at	Creator	Custom.Label	Data.Source	Graph	Table
001	2021-09-19/11:43	decill	Cancer Types by Year	OGD_krebs_ext_KREBS_1	NULL	175x3
002	2021-09-19/14:17	meindl	Household Forecast	OGD_f1741_HH_Proj_1	Timeseries	NULL
003	2021-09-20/06:34	decill	Structure of Earnings	OGD_veste301_Veste301_1	Barplot	122 x 7
004	2021-09-21/13:29	kowa	Tourism: Correlations	detouextregsai	Scatterplot	02 x 9

Each row represents a **graph**, a **table** or both. Notice the `NULL` values in the last two rows

The columns **Creator** and **Data Source** reference the user and dataset behind a component

Single items can be requested with `/GET table` and `/GET graph`. They return a **rendered** version of a component for the **CMS**

The json payload of the response contains a `<div>` or `<table>` tag and a `<script>` tag which define the component

```
# get the component from the api
response <- httr::GET(
  "rsconnect.local/component-api/graph?Key=001")
payload <- httr::content(response)

# fill a html template
fill_table_template(
  # pass a javascript string
  script = payload['script'],
  # pass container (<div>) as html string
  container = payload['container']
)
```

The **template** defines where the component (graph) is inserted and loads all necessary **js/css dependencies**

```
<head>
  <script src="highcharts.js"/>
  <script src="rsconnect.local/component-api/utils.js"/>
</head>

<body>
  <h1>Graph template</h1>
  <!-- define placeholder container for the graph -->
  ${container}

  <!-- fill container using a script tag -->
  ${script}
</body>
```

**utils.js** is a helper package that defines functions for **layouting** of components, **formatting tooltips**, etc.

## Frameworks

- **shiny** defines the component builder **app**
- The component **API** is generated with **plumber**

## Front End

- **highcharter** builds the **graphs**
- All **tables** are generated using **datatables.js**

## Back End

- **httr** communicates with the **plumber API** and the **STATcube API**
- Data **import** is performed with the **STATcubeR** package

Currently, the package **STATcubeR** is available on github. It will be released to **CRAN** when the **STATcube** REST API becomes available for external users

The internal packages **STATgraph** and **STATtable**, which create graphs and tables based on the **STATcubeR** data interface, might become open-source once the pipeline is fully integrated in the **CMS** workflow

Thank you for your attention

More information as well as the source code for these slides can be found at <https://github.com/GregorDeCillia/pipedream>

A video with this presentation is available at <https://youtu.be/kjUtApjK6XE?t=215>