# Goal

We would like to see a SPA with two pages (routes): a details page for the TV show "Powerpuff Girls", and an episode detail page with information about specific episodes. Note that since this is a SPA, there should be no full-page refresh when navigating between those routes.

The TV show page should **at least** display the following information (you're free to add more):

* Show title
* Show Description
* Show cover image
* Episode list. Every episode in the list should link to a details page for that specific episode

The episode detail page should contain at least:

* Episode title
* Episode Summary
* Episode cover image

We expect your application to have at least a page for one specific show: "The Powerpuff Girls". You are free to make other shows available as well.

As an example of how a TV show page might look, you can look at the TV Maze show pages here: <http://www.tvmaze.com/shows/6771/the-powerpuff-girls>.

# Data

Data in your application should be retrieved from the TV Maze API (<http://www.tvmaze.com/api>). TV Maze provides a free Rest API with data that strongly resembles the data we use internally to render our VOD platforms like Videoland and RTL XL. We expect that you use some kind of state management in you app. We encourage you to use an existing state management library, like Redux.

# Design

The "look and feel" (design) of the application is completely up to you. Rest assured, we will not judge you on your design skills. But there are some things we expect to see:

* Use CSS to style your components. You're encouraged to use preprocessor such as Sass or Less.
* The site should be responsive and should be usable both on desktop and (small) mobile screens. Define at least one breakpoint in your CSS.

# Deliverables

**Try to spend a maximum of 4-6 hours on this assignment. You are free to use more time, but we don't expect you to.**

Check in all your source code into Git repository. Either in git.epam.com or github.com.

Use branching at your discretion, but the end result has to be available in the master branch.

Your app should run from local web server (localhost). After checkout of the code, the only things we should do in order to view the end result is:

* Install dependencies
* Start local web server
* Open web browser

Documentation, especially on building and running the app, is appreciated. It is up to you to decide how comprehensive this documentation should be.

# Other considerations

* Your application should work in all modern browsers.
* We prefer code written in ES6 (use a transpiler for browser compatibility).
* English is the preferred language in code comments, documents, variable names, etc.
* Preferably use saga for side effects.

In general, choose a development methodology and tool set you feel most comfortable with, as long as you adhere to the requirements below.

# Non-functional requirements

**!IMPORTANT!** The main goal of this exercise not only to build working app, but also to show all your knowledge of best practices, technical skills and affords. Please provide clean, well structured, extendable (imagine that this is only first user story of a long project), maintainable solution. Don’t forget to cover your code with unit/snapshot tests as much as you can. Also please keep in mind that code should be readable and self-explaining. All this points will be taken into account in final results.

# Summary of all requirements

* Use **React** to render UI components.
* Setup **two routes** in you app, one for TV shows and one for episodes.
* Retrieve data from **TV Maze REST API**
* Setup **state management** and **data flow**
* Style your components with **CSS**, make sure that the end result is **responsive** and has **at least one breakpoint.**
* All code should be **checked in into a Git repository**
* Unit/snapshot tests.
* Non-functional requirements.
* **Documentation is optional but appreciated**, the **same goes for comments in your code**