# Exercises: AJAX with Promises

Problems for exercises and homework for the [“JavaScript Applications” course @ SoftUni](https://softuni.bg/courses/javascript-applications). Submit your solutions in the SoftUni judge system at <https://judge.softuni.bg/Contests/XXX/>.

**Exercises: AJAX with promises. CRUD with GitHub Issues**

## <First Problem>

Your task is to implement a Web page that …

### Screenshots

…

…

### HTML and JavaScript Code

You are given the following **HTML** code:

|  |
| --- |
| index.html |
| <!DOCTYPE html>  <html>  <head>  <script src="**scripts.js**"></script>  </head>  …  </html> |

It comes together with the following **JavaScript** code:

|  |
| --- |
| scripts.js |
| function XXX() { … } |

Implement the above **function** and return a **JS object** that provides the following functionality:

* …
* …
* …

Submit in the judge the JS code (implementation) of the above function. It may hold other functions in its body.

### Hints

* …
* …
* …

## Students

Your task is to create functionality for creating and listing students from a database in Kinvey.

Here are the Kinvey application key and secret:

* APP KEY: kid\_BJXTsSi-e
* APP SECRET: 447b8e7046f048039d95610c1b039390

Here is also a test user for you to log in with:

* Username: guest
* Password: guest

There is collection called students there, in the application.

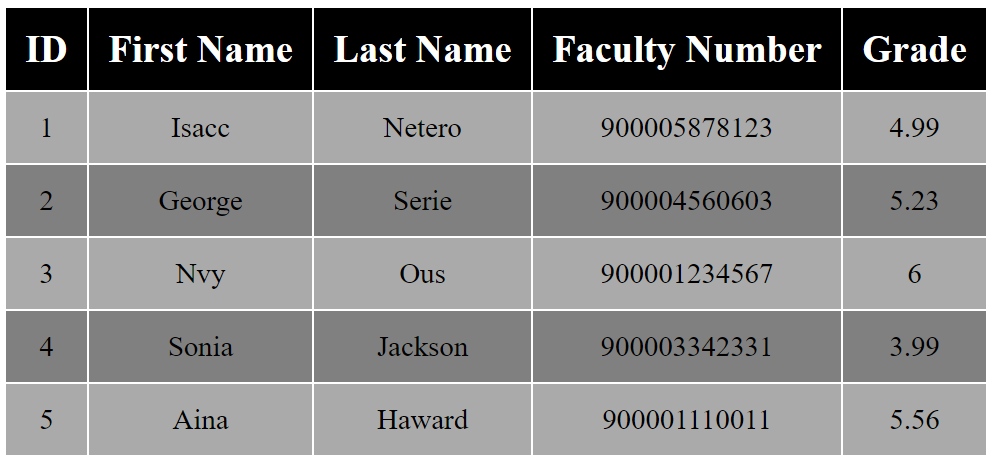
The student entity has:

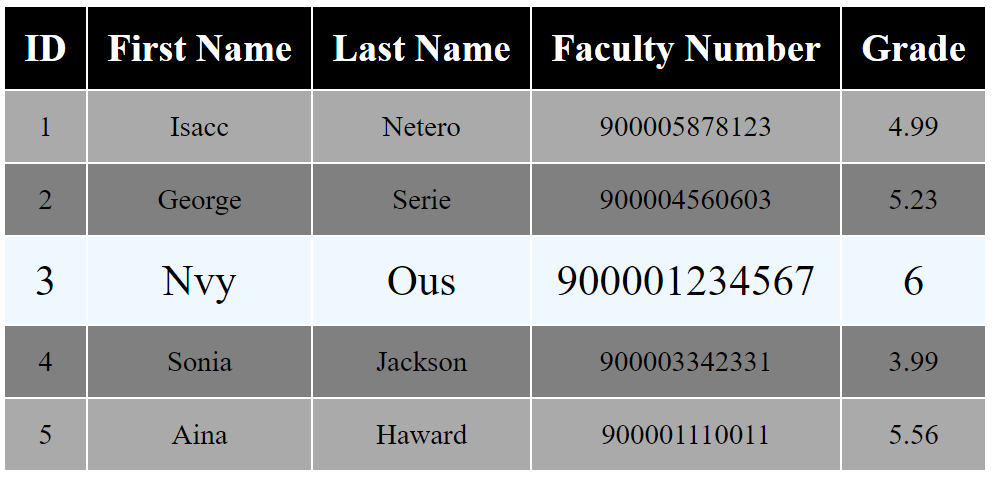
* ID – a number, **non-empty**
* FirstName – a String, **non-empty**
* LastName – a String, **non-empty**
* FacultyNumber – a String of **numbers**, **non-empty**
* Grade – a number, **non-empty**

You need to write functionality for creating students. When you are creating a new student, make sure you name these properties perfectly. Create at least one student to test your code.

You will also need to extract the students. You will be given an **HTML template** with a table in it. Create an AJAX request which extracts all the students. Upon fetching all the students from Kinvey, add them to the table each on a new row, **sorted** **ascendingly** by ID.

### Screenshots





## <Third Problem>

Your task is to implement a Web page that …

### Screenshots

…

…

### HTML and JavaScript Code

You are given the following **HTML** code:

|  |
| --- |
| index.html |
| <!DOCTYPE html>  <html>  <head>  <script src="**scripts.js**"></script>  </head>  …  </html> |

It comes together with the following **JavaScript** code:

|  |
| --- |
| scripts.js |
| function XXX() { … } |

Implement the above **function** and return a **JS object** that provides the following functionality:

* …
* …
* …

Submit in the judge the JS code (implementation) of the above function. It may hold other functions in its body.

### Hints

* …
* …
* …

## Countries & Towns\*

### Create a Backend at kinvey.com

Register at **kinvey.com**, create an application and two classes: **Country(name)** and **Town(name, country)**. Fill sample data for further use. Submit in the homework screenshots of your classes from kinvey.com.

### List Countries

Create a JavaScript application (HTML + CSS + JS + jQuery) that loads and displays all countries from your application at Parse.com into a HTML page.

### Edit Countries

Extend the previous application with add / edit / delete for countries.

### List Towns by Country

Extend the previous application to show all towns when a certain country is selected.

### CRUD Towns

Extend the previous application to implement add / edit / delete for towns.