

More Exercises: Remote Databases

For this exercise you have to create a new application at kinvey.com. And for each of the following tasks you must create a **different collection**.

1. Wild Wild West

Write **REST services** for a simple Western game. Create a collection **player** (**name**, **money**, **bullets**) to hold information about the players in the game.

- **name** - **string** representing the name of the current player.
- **money** - **integer number** representing the current player's money.
- **bullets** - **integer number** representing the current bullets of the player.

HTML and JS

You will be provided with a skeleton project containing an **HTML template** and some **JS files**. The **loadCanvas.js** is a simple implementation for the game and your job is to **attach events** to all the buttons and make the needed **AJAX requests**.

When the page is loaded a **GET** request should be sent to the server to get all players and load them in the **div** with **ID players**. An example entry is left in the HTML to demonstrate the representation of a player and their placement.

Whenever the [**Save**] button is pressed, the progress of the current player (if any) should be saved (a **PUT** request sent to the server with the new data):

- The **canvas** and buttons [**Save**] and [**Reload**] should be **hidden**
- The **clearInterval** should be called on the **canvas.intervalId** property (used for the main loop of the game)

Whenever the [**Reload**] button is pressed, the player's money should be **reduced by 60** and their bullets should be **set to 6**.

Whenever the [**Add Player**] button is clicked, a new Player with the name specified in the corresponding input should be created and the players should be reloaded to display the new entry. Each new player **starts with 500 Money and 6 bullets**.

Pressing the [**Play**] button on a player should:

- Call the [**Save**] button
- Display the **canvas**, [**Save**] and [**Reload**] buttons
- Call the **loadCanvas()** function (from the **loadCanvas.js**)
- Pass to it the **new player** as an **object** (containing properties **name**, **money** and **bullets**)

When a player's [**Delete**] button is pressed, the player should be deleted (both from the HTML and from the server).

Examples

Players

Name:George	Name:Nick
Money:500	Money:500
Bullets:6	Bullets:6
<button>Play</button>	<button>Play</button>
<button>Delete</button>	<button>Delete</button>

Add Player

Add Player

