

„Any fool can write code that a computer can understand. Good programmers write code that humans can understand.“

-Martin Fowler



## Objective-C Programming Task

Your task is to create a simple iOS app, that provides UI for entering user credentials, a list of available game worlds and network handling to a server. The goal is to give you the chance to give us an impression of your coding style, coding skills and architectural knowledge. The task should not just be solved, but solved in a beautiful way - simply impress us 😊. Keep the code maintainable and extendable.

The app should be written in Objective-C and run on iPhone and iPad devices. All the program logic has to be written by yourself. The use of libraries etc., of course, is fine where appropriate, but please document where you got your results from. If possible document your work progress with commits to your code repository. It's **not allowed** to publish the result. Feel free to email us if any questions arise.

The goal of the app is to display a list of available game worlds for a specified user. The user should be able to enter his email address and corresponding password. By tapping a button the user should be able to request his available game worlds. Display the names of available game worlds in a list to the user.

You need to send a POST request to <https://backend1.lordsandknights.com/XYRALITY/WebObjects/BKLoginServer.woa/wa/worlds> to request the available game worlds. The documentation for this action is attached at the end of this file.

Create the string for the field *deviceType* like this `[NSString stringWithFormat:@"%@@ - %@%@@", [[UIDevice currentDevice] model], [[UIDevice currentDevice] systemName], [[UIDevice currentDevice] systemVersion]]`;

You can use `[[NSUUID UUID] UUIDString]`; to create a unique device identifier.

Use [ios.test@xyrality.com](mailto:ios.test@xyrality.com) for login parameter and *password* for password parameter.

The response you will get when calling the action is in plist format.

---

## DirectAction/worlds

This request provides you with the list of available game worlds.

### Parameters

login	The device identifier or email address used to sign in.
password	The corresponding password for the login.
deviceType	A string describing the hardware and operating system of a device.
deviceId	A unique identifier that identifies the device.

### Return Value

The response is a dictionary containing an array (with the field `allAvailableWorlds`) with all available game worlds. The description of a world is represented in a dictionary. Each world has the following description structure:

```
country = ISO country code string
id = a unique identifier
language = ISO language code string
mapURL = A URL to a map server
name = Name string of the world
url = A URL to the world server
worldStatus = {
    description = Description string of a status
    id = Identifier indicating the status
}
```

### Example response

```
{
  allAvailableWorlds = (
    {
      country = AU;
      id = 126;
      language = en;
      mapURL = "http://maps2.lordsandknights.com/v2/LKWorldServer-AU-6";
      name = "Australia 6 (recommended)";
      url = "http://backend2.lordsandknights.com/XYRALITY/WebObjects/LKWorldServer-AU-6.woa";
      worldStatus = {
        description = online;
        id = 3;
      };
    }
  );
}
```