Recrutement Keyrus

Tests Techniques

28/07/2017

Historique du document

Version	Date	Objet de la version	Auteur	Relecteurs
1.0	28/07/2017	Version initiale.	Gabriel Bremont	Thibaut Varlez, Tamara Kondakjian

Sommaire :

1.Introduction

This document describes a test to validate your knowledge of basic but critical build of an API and its website. Please send us all the source code within GIT repository (a public or private GIT repository of you own, or a zip of a local repository).

2.Purpose

Here you are going to create a simple address book website with the correlated API.

This will have all the common functionalities of an address book like list, add and modify contacts.

3.Discover the MODEL

Complete description of the expected model to use for the API:

```
## Contact
{
	id = " 4OGDGgJObh2r6dQPjZ6Hqjx+fB6Y7nUEsP58huerheo=",
	firstName = "Steve",
	lastName = "Jobs",
	birthDate = "1955/02/24",
	company = "Apple",
	emails =
	[
		{
			email = "steve@apple.com",
			type = "PRO"
		}
	],
	addresses =
	[
		{
			streetNb = 1,
			streetType = "ROAD",
			street = "Infinity Loop",
			city = "Cupertino",
			state = "CA",
			zipCode = 94014,
			country = "United States of America",
			type = "PRO"
		}
	]
}
```

At least one of the following field must be filed: firstName, lastName or company. The id field is generated server side and can't be modify by the client.

4.Projects

Create the projects you need with the dependencies of you choice to create the CRUD API on this object and a website that use this API. You may use any language, framework, database and library as long as they are production-ready.

5.Backend - The API

Expectations on the API:

- List all the contacts.
- Retrieve a specific contact.
- Add a new contact.
- Modify an existing contact.

6.Frontend - First view: list of contacts

Create a view that queries the API you made for the list of contacts. For each contact, display the ***firstName***, ***lastName*** and ***company***.

Notice that only one of these fields is mandatory.

7.Frontend - Second view: contact content

When a contact is selected from the first view, display a view that contains all the fields of a contact.

Add a button to make this view into editing/validate state: all fields could be changed; emails and addresses can be changed, added or deleted.

8.Extra credit

- Add a button in the list of contacts to add a new contact.
- Suggest a solution to change the way the list is sorted.
- Suggest a way to easily but safely delete a contact in the list.

- Use engaging transitions between the list of contacts and the contact content
- Find an easy way to navigate from contact to contact in the contact content view.