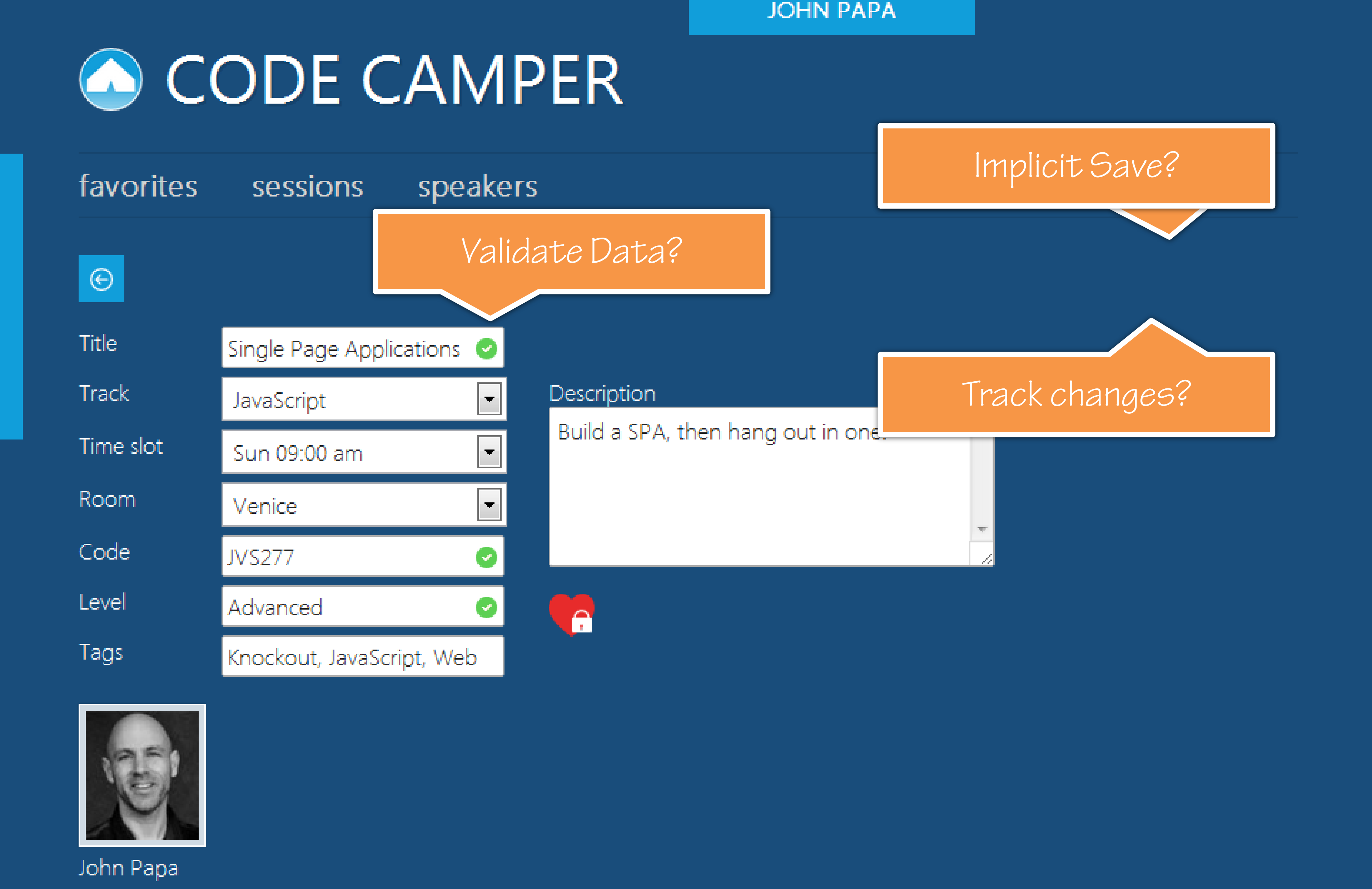
Saving, Change Tracking and Validation

John Papa

Twitter: @john_papa

Saving Data - Agenda

- Change Tracking
- Asynchronous Commands
- Saving Data
- Validation



Title	Single Page Applications
Track	JavaScript
Time slot	Sun 09:00 am
Room	Venice
Code	JVS277
Level	Advanced
Tags	Knockout, JavaScript, Web

Description

Build a SPA, then hang out in one.

John Papa

Implicit Save?

Validate Data?

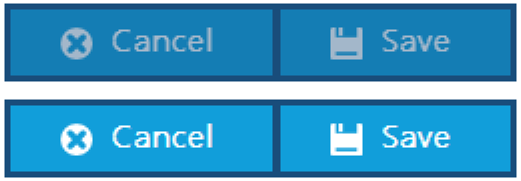
Track changes?

Saving Data - Agenda

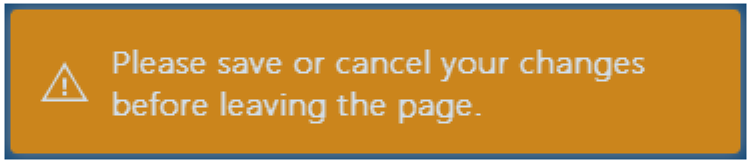
- Change Tracking
- Asynchronous Commands
- Saving Data
- Validation

Why Track Changes?

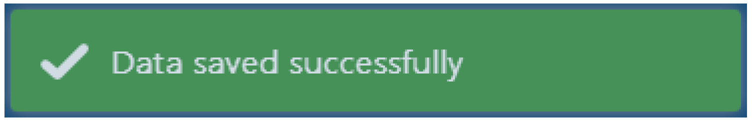
- **Enabling / Disabling of Buttons**



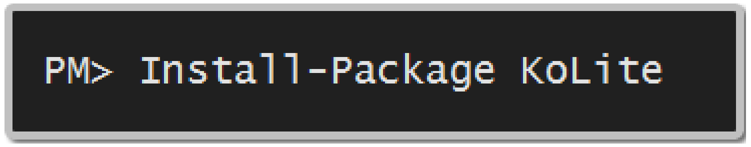
- **Prevent Navigation**


Please save or cancel your changes before leaving the page.

- **Make sure the user is done**


Data saved successfully

- **KoLite**
  - ko.dirtyFlag
  - https://nuget.org/packages/KoLite



```
PM> Install-Package KoLite
```

Saving Data - Agenda

- Change Tracking
- Asynchronous Commands
- Saving Data
- Validation

Asynchronous Commands

Disabled and Activity

ko.activity

ko.asyncCommand

ko.DirtyFlag

```
PM> Install-Package KoLite
```

KoLite on NuGet

Defining a ko.asyncCommand

```
saveCmd = ko.asyncCommand({
    execute: function(complete) {
        $.when(datacontext.persons.updateData(speaker()))
            .always(complete);
    },
    canExecute: function(isExecuting) {
        return !isExecuting && isDirty() && isValid();
    }
})
```

1 Define the command

2 The Function to Execute

3 Save the Data

4 Define the Conditions When the User Can Save

Binding a ko.asyncCommand

```
<button data-bind="command: saveCmd, activity: saveCmd.isExecuting">
Save</button>
```

```
saveCmd = ko.asyncCommand({
    execute: function(complete) {
        $.when(datacontext.persons.updateData(speaker()))
            .always(complete);
    },
    canExecute: function(isExecuting) {
        return !isExecuting && isDirty() && isValid();
    }
})
```

Saving Data - Agenda

- Change Tracking
- Asynchronous Commands
- Saving Data
- Validation

Defining Model Validation

Common Rules

```
self.firstName = ko.observable().extend({ required: true });

self.email = ko.observable().extend({ email: true });

self.blog = ko.observable().extend({
        pattern: {
            message: 'Not a valid url',
            params: /[@]([A-Za-z0-9_]{1,15})/i
        }
    });
```

Built in Validation Rule

Custom Validation Rule

Knockout Validation Plug-In

- Define validation on observables
- Configure behavior
- Check validation errors

```
PM> Install-Package Knockout.Validation
```

http://nuget.org/packages/Knockout.Validation

Saving Data - Recap

- **Saving Data**
  - datacontext to dataservice to Web API

```
PM> Install-Package KoLite
```

- **KoLite**
  - Change Tracking
  - Asynchronous Commands
  - Command Activity
  - https://nuget.org/packages/KoLite

- **Validation**
  - HTML5 and JavaScript based
  - Knockout.Validation
    - http://nuget.org/packages/Knockout.Validation

```
PM> Install-Package Knockout.Validation
```