JSON Forms 2.0
Form-based UIs for Web Applications

Maximilian Koegel
mkoegel@eclipsesource.com
Applications frequently require forms

<table>
<thead>
<tr>
<th>Name*</th>
<th>Send email to Adrian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Due Date</td>
<td>Rating</td>
</tr>
<tr>
<td>Description</td>
<td>3</td>
</tr>
</tbody>
</table>

Confirm if you have passed the subject
Hereby ...

<table>
<thead>
<tr>
<th>Recurrence</th>
<th>Recurrence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily</td>
<td></td>
</tr>
</tbody>
</table>
Forms map data into a UI

![Task form](image)

- name : EString
- description : EString
- dueDate : EDate
- done : EBoolean = false

**Name**

Send email to Adrian

<table>
<thead>
<tr>
<th>Due Date</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

**Description**

Confirm if you have passed the subject

**Recurrence**

Daily
Declarative Data Definition = JSON Schema

- “XML Schema for JSON”
- Format to describe JSON-based data
- Focused on data validation
- Can be created manually
- Can be generated from EMF Ecore

→ Used in JSON Forms to define the data
Declarative UI Definition = UI Schema

- Declarative format for form-based UIs
- Describes the layout of a form
- Describes the data binding
- Does **not** describe the data
- References the data schema (JSON Schema)

→ Used in JSON Forms to define the form-based UI
Combining JSON and UI Schema = Rendering

```
1. {  
2.   "type": "object",  
3.   "properties": {  
4.     "name": {  
5.       "type": "string",  
6.       "minLength": 1  
7.     },  
8.     "description": {  
9.       "type": "string"  
10.    },  
11.   "done": {  
12.     "type": "boolean"  
13.    },  
14.   "due_date": {  
15.     "type": "string",  
16.     "format": "date"  
17.    },  
18.   "rating": {  
19.     "type": "integer",  
20.     "maximum": 5  
21.    },  
22.   "recurrence": {  
23.     "type": "string",  
24.     "enum": [  
25.       "Never",  
26.       "Daily",  
27.       "Weekly",  
28.       "Monthly"  
29.     ]  
30.   }  
31. },  
32. })

1. s{  
2.   "type": "VerticalLayout",  
3.   "elements": [  
4.     {  
5.       "type": "Control",  
6.       "label": false,  
7.       "scope": "#/properties/done"  
8.     },  
9.     {  
10.       "type": "Control",  
11.       "scope": "#/properties/name"  
12.     },  
13.     {  
14.       "type": "Control",  
15.       "scope": "#/properties/due_date"  
16.     },  
17.     {  
18.       "type": "Control",  
19.       "scope": "#/properties/rating"  
20.     }  
21.   ]  
22. },  
23. })
```

**JSON Schema**
(Data Schema)

**UI Schema**

**User Interface**

Prepare slides for demo

<table>
<thead>
<tr>
<th>Due Date</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018-03-27</td>
<td>8</td>
</tr>
</tbody>
</table>

should be <= 5

Description
This is some awesome multi-line text to show that it may or may not make sense to display it this way

Recurrence

<table>
<thead>
<tr>
<th>Recurrence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Never</td>
</tr>
</tbody>
</table>
Rendering Controls

JSON Schema (Data Schema)  UI Schema  User Interface

Task Form

Name

Description

Done?

Name*

Prepare slides for demo

Due Date

2018-03-27

Rating

8

should be <= 5

Description

This is some awesome multiline text to show that it may or may not make sense to display it this way

Recurrence

Never

Recurrence Interval
Rendering Controls

JSON Schema (Data Schema) + UI Schema = User Interface

Data Binding

Task Form

Name

Description

Done?

Name*
Prepare slides for demo

Due Date
2018-03-27

Rating
8

should be <= 5

Description
This is some awesome multi-line text to show that it may or may not make sense to display it this way

Recurrence
Never

Recurrence Interval
Rendering Layouts

JSON Schema (Data Schema)

UI Schema

User Interface
Rendering Layouts
Demo!

Welcome to JSON Forms with React
More Forms, Less Code.

Bound data

```json
{
  "name": "Send email to Adrian",
  "description": "Confirm if you have passed the subject\n  Herby ...",
  "done": true,
  "recurrence": "Daily",
  "rating": 3
}
```

Name*
Send email to Adrian

<table>
<thead>
<tr>
<th>Due Date</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

Description:
Confirm if you have passed the subject
Herby ...

Recurrence
Daily

Recurrence Interval
Demo: Validation

Name* should NOT be shorter than 1 characters

Due Date
2018-03-27

Rating
8

Description
This is some awesome multi-line text to show that it may or may not make sense to display it this way

Recurrence
Never

Recurrence Interval
Demo: Visibility Rules
Demo: Custom Renderers

Rating
3
Demo: EMF Forms Integration

EMF Forms

Ecore + View Model = User Interface

JSON Schema Export

UI Schema Export

JSON Forms

JSON Schema

UI Schema = User Interface

Web Renderer Set

SWT/JavaFX/RAP/Tabris Renderer Sets
Demo: Mobile Renderer with Ionic
What's new with 2.0? (1/2)

- **Improved Architecture**
  - Redux-based architecture
  - Renderers are simpler (no update handling)
- **Renderer Sets**
  - Material React and Vanilla React
  - Ionic (WIP)
  - Angular Material (WIP)
  - Easy to create new one: 20-50 days
What's new with 2.0? (2/2)

- Improved Schema support:
  - Any of & All of
  - Reference resolving
- Support for Trees:
  - TreeWithDetail
  - Tree-Editor
- Eclipse Theia Integration: Theia-Tree-Editor
More Information

JSON Forms
Web: http://jsonforms.io
Twitter: #jsonforms @JSONForms

EMF Forms
Web: http://emfforms.org
Twitter: #emfforms @EMFFForms

EclipseCon Europe
Building Web-based Modeling Tools based on Theia - 11:30 AM - Theater Stage
The Busy RCP Developer’s Guide to Eclipse Theia - 1:15 PM - Theater Stage
Evaluate this Session

Sign in and vote at eclipsecon.org

WITH

-1 0 +1