

JavaScript Front-End Web App Tutorial Part 2: Adding Constraint Validation

**Learn how to build a front-end web
application with responsive constraint
validation using plain JavaScript**

Gerd Wagner <G.Wagner@b-tu.de>

JavaScript Front-End Web App Tutorial Part 2: Adding Constraint Validation: Learn how to build a front-end web application with responsive constraint validation using plain JavaScript

by Gerd Wagner

Warning: This tutorial manuscript may contain errors and may still be incomplete. Please report any issue to Gerd Wagner at G.Wagner@b-tu.de.

This tutorial is also available in the following formats: PDF [[validation-tutorial.pdf](#)]. You may run the example app [[ValidationApp/index.html](#)] from our server, or download the code [[ValidationApp.zip](#)] as a ZIP archive file. See also our Web Engineering project page [<http://web-engineering.info/>].

Publication date 2015-11-12

Copyright © 2014-2015 Gerd Wagner

This tutorial article, along with any associated source code, is licensed under The Code Project Open License (CPOL) [<http://www.codeproject.com/info/cpol10.aspx>], implying that the associated code is provided "as-is", can be modified to create derivative works, can be redistributed, and can be used in commercial applications, but the article must not be distributed or republished without the author's consent.

Table of Contents

Foreword	vi
1. Integrity Constraints and Data Validation	1
1. Introduction	1
2. Integrity Constraints	1
2.1. String Length Constraints	3
2.2. Mandatory Value Constraints	3
2.3. Range Constraints	4
2.4. Interval Constraints	5
2.5. Pattern Constraints	5
2.6. Cardinality Constraints	6
2.7. Uniqueness Constraints	7
2.8. Standard Identifiers (Primary Keys)	7
2.9. Referential Integrity Constraints	8
2.10. Frozen Value Constraints	8
2.11. Beyond property constraints	8
3. Responsive Validation	9
4. Constraint Validation in MVC Applications	9
5. Criteria for Evaluating the Validation Support of Frameworks	10
2. Constraint Validation in a Plain JS Front-End App	12
1. Introduction	12
2. Using the HTML5 Form Validation API	13
3. New Issues	13
4. Make a JavaScript Data Model	14
5. Set up the Folder Structure with Library Files	15
5.1. Style the user interface with CSS	15
5.2. Provide general utility functions and JavaScript fixes in library files	15
5.3. Create a start page	16
6. Write the Model Code	16
6.1. Summary	16
6.2. Encode the model class as a constructor function	17
6.3. Encode the property checks	18
6.4. Encode the property setters	18
6.5. Add a serialization function	19
6.6. Data management operations	19
7. The View and Controller Layers	20
7.1. The data management UI pages	21
7.2. Initialize the app	22
7.3. Initialize the data management use cases	22
7.4. Set up the user interface	23
8. Run the App and Get the Code	26
9. Evaluation	26
10. Possible Variations and Extensions	26
10.1. Adding an object-level custom validation function	26
10.2. Simplifying forms with implicit labels	26
11. Points of Attention	27
11.1. Database size and memory management	27
11.2. Boilerplate code	27
12. Practice Project	27

List of Figures

1.1. The object type <code>Person</code> with an interval constraint	5
1.2. The object type <code>Book</code> with a pattern constraint	6
1.3. Two object types with cardinality constraints	6
1.4. The object type <code>Book</code> with a uniqueness constraint	7
1.5. The object type <code>Book</code> with a standard identifier declaration	8
2.1. A platform-independent design model with the object type <code>Book</code> and two invariants	12
2.2. Deriving a JavaScript data model from an information design model	14
2.3. The validation app's start page <code>index.html</code>	16
2.4. The object type <code>Movie</code> defined with several constraints.	28

List of Tables

2.1. Sample data for <code>Book</code>	13
2.2. Datatype mapping	17
2.3. Evaluation	26
2.4. Sample data	28

[Click here to download full PDF material](#)