APPLICATION DEVELOPMENT USING METEOR FRAMEWORK

Victor LUCA, student FAF-141; Mihail KULEV, assoc. prof. PhD

Technical University of Moldova

Abstract: *In the paper a full stack open source framework for creating web and mobile applications in pure JavaScript, called Meteor, has been presented. Principles, components, live HTML templates, livequery, security and deploying of the framework Meteor have been described. Framework Meteor integrates with MongoDB, uses the Distributed Data Protocol, allows very rapid prototyping and produces cross-platform (web, Android, iOS) code.*

Keywords: *Meteor framework, application development, MongoDB, Distributed Data Protocol, web, Android, iOS, cross-platform*

1. Introduction

Meteor – is a full stack open source framework for building web and mobile applications in pure JavaScript. Meteor allows for very rapid prototyping and produces cross-platform (web, Android, iOS) code. It integrates tightly with MongoDB and uses the Distributed Data Protocol and a public–subscribe pattern to automatically propagate data changes to clients in real-time without requiring the developer to write any synchronization code.

2. Principles of framework

1. Data on the Wire - sends HTML over the network. The server sends data and lets the client render it.
2. One Language - allows you write both the client and the server parts of your application in JavaScript.
3. Database Everywhere - using the same methods to access your database from the client or the server.
4. Latency Compensation - prefetches data and simulates models to make it look like server method calls return instantly.
5. Full Stack Reactivity - real-time is the default. All layers, from database to template, update themselves automatically when necessary.
6. Embrace the Ecosystem - is open source and integrates with existing open source tools and frameworks.

3. Components

Meteor has two main components:

1. Library packages – it contains more than 8000 packages which suits needs of the developers.
2. Command line – allows installing, removing packages as well as building application and running it in different environments.

4. Live HTML templates

Meteor is using by default Blaze – a powerful library for creating live-updating user interfaces. Basically it is plane HTML which is rendered on the client, but it's real time and changes made in database propagated and automatically re-rendered on the client. Since Meteor supports most of the template languages it allows developers to decide which one is more suitable for them. In latest versions meteor allows using such front-end frameworks as: Angular, Backbone or React. For example, if we have the following template:

```
<template name="variants">
 <ul>
  {{#each this.variants}}
   <li>{{text}}</li>
  {{/each}}
  </ul>
 </template>
```

Template receives object throw which function inside of template iterates and inserts elements in the unordered list.

5. Livequery

Meteor Livequery is a family of live database connectors. These connectors perform “live query” to your Mongo database. They return not only result of the query, but also they create a stream throw which data is propagated and is automatically updated in case changes were made. An example of the query is:

```
Template.adminDashboard.events({
  'click .givePoints': function () {
    Players.update(Session.get("currentPlayer"), {$inc: {score: 5}});
  }
})
```

This query is fired on the client side when click event happen and it updates the currently selected player by increasing its score with 5 points.

6. Security

In meteor the same database API is shared by the client and server, but while code is running on the server it has a direct access to the database, code on the client does not have direct access. Client make changes in database throw methods which are called on the server to perform changes in database. Meteor doesn't allow db direct injection. For storing hashes and passwords is used bcrypt which is considered one of the best key derivation function.

7. Deploying

How can we check if our application is good or not? The answer is simple: “Give it to users!”. Meteor framework takes care of deploying your application and even give you a server to run it on. Meteor designed its own infrastructure for running prototypes of applications. Users can access in real time application by going to one of the sub-servers of Meteor.

Bibliography

1. METEOR. https://www.meteor.com/
2. METEOR. Tutorial. https://www.meteor.com/tutorials/blaze/creating-an-app
3. Tom Coleman, Sacha Greif *Discover Meteor*. https://www.discovermeteor.com/
4. The MongoDB 3.2 Manual. https://docs.mongodb.org/manual/
5. Meteor forums. https://forums.meteor.com/
6. Welcome to Practical Meteor! http://practical.meteor.com/super-easy-deploy