INTEGER TO ROMAN NUMERALS CONVERTER

For this task I chose **Angular.js** as main because of its clearness and visibility in projects like that.

First of all I've created a simple view of application with using **Bootstrap 3** which contains header and main container with input field and block for results.

*Config* with single route for angular app was created for simplicity of view and flexibility in future growth of application.

*ConverterFactory* provides controller with functions for every necessary calculations which allows to keep *mainController* thin. These functions are:  
 - *checkBorders(number)* — returns *false* if current number is out of borders or doesn’t exist in otherwise returns *true*.  
 - *getBordersInfo()* — returns *BORDERS\_INFO* which contains information about borders based on *MIN\_VALUE* and *MAX\_VALUE* constants.  
 - *romanNumeralConverter(number)* — returns converted integer into Roman numeral.

As we know, roman numbers are formed by combining symbols and adding the values, so II is two (two ones) and XIII is thirteen (a ten and three ones). Because of this “Summand-combined model” I decided to build main function as recursive because this shape fits there pretty well. I created two twin arrays: *DECIMAL* with arabic numbers and *ROMAN* with the roman characters both sorted in descending order. Therefore *for* loop is looking for biggest included summand, decreases number for the value found and returns part of roman number and then calls new *romanNumeralConverter(number)* with decreased value as argument. Last iteration with number less than *MIN\_VALUE* will return empty string so calling recursive function will stop and first called function will return result which is ready-to-use.

Tests are checking for correct results and including in borders. They were written with help of *Jasmine* and passed successfully.