Prepare system environment

To auto detect the right stage and the according configuration, the voting application relies on a system environment parameter.

At the moment four different stages are defined:

* development: Runs on a single MongoDB instance and requires JWT based authorization for the usage of the REST interfaces.
* development.no.security: Runs on a single MongoDB instance but doesn’t require any authorization for the usage of the REST interfaces.
* quality.assurance: Runs on a MongoDB replication set as database and requires JWT based authorization for the usage of the REST interfaces.
* production: Runs on a MongoDB replication set as database and requires JWT based authorization for the usage of the REST interfaces.

The different stages also differ in granularity and handling of log output and strength of preconfigured passwords.

The expected system variable is called “voting.stage”. If the variable is not present or has an invalid value, the application will not start.

To set the value in Windows, type the following command in the command prompt or PowerShell  
> setx voting.stage development

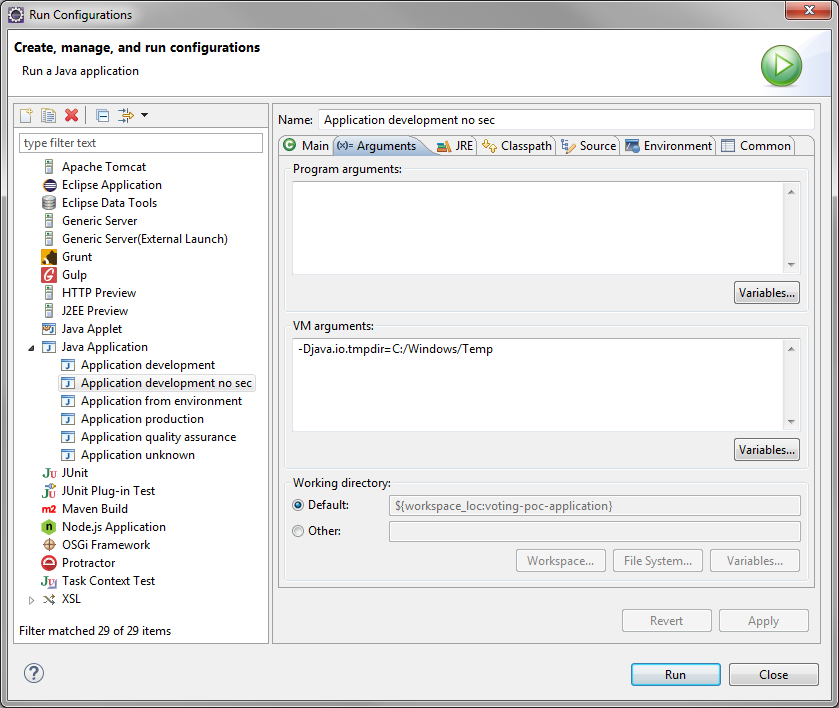
To set the value in Windows, type the following command in the command prompt or PowerShell  
> setx voting.stage development

Linux and Mac user know how to set an environment variable.

Setup Eclipse

Known issues with Spring Profiles, Eclipse and JRE configuration

For some not yet identified reasons, when the application profile is set via Eclipse Run Configurations, it is possible that the “temp” folder of Windows systems directs to a protected system folder. If the error occurs on your machine, add the temp directory as JRE parameter in Eclipse.



Note:  
When a new profile is introduced the “logback-spring.xml” file must be adapted. Otherwise no log output is shown in IDE or printed to log file.

Setup Database

Version

The Voting PoC application is developed and runs with release 3.2.4 (April 2016) of MongoDB. The installer /packages are available on [<https://www.mongodb.com/download-center#community>].

Installation

There is a very good installation guide for MongoDB available on [<https://docs.mongodb.org/manual/installation>]. Therefore this guide will concentrate onto the application specific installation and setup.

For the ease of usage, define the path to the MongoDB binaries/executables in the environment variables.

Installation of development environment

The installation database is a single instance MongoDB. The application is designed to work with authentication even on development level. To keep the installation as simple as easy, default scripts for the basic setup are provided.

Prepare the file system to host the MongoDB

On the root directory create a directory structure like shown in the picture. Place the initialization scripts from the tools repository in the “dev” directory.

Setup the database

1. Start the MongoDB via command prompt  
   > mongod –config init\_mongo.conf  
   Check if the start was successful via “log/database/init.log
2. Open a new command prompt and execute the js file via mongo shell   
   > mongo 01\_prepare\_database.js
3. Shut down the MongoDB instance
4. Restart the the database with the other configuration file  
   > mongod –config mongo.conf  
   Check if the start was successful via “log/database/node.log
5. Execute the following js scripts via mongo shell   
   > mongo 02\_init\_database\_user.js  
   > mongo 03\_init\_database.js
6. Import an initial data set with mongoimport (optional)  
   > mongoimport -u voting\_app -p 12345678 --authenticationDatabase application\_users -d voting\_poc -c users < users.json  
   > mongoimport -u voting\_app -p 12345678 --authenticationDatabase application\_users -d voting\_poc -c votes < votes.json

MongoDB is now ready to be used in development mode.

Note:  
If your directory structure is differing from the proposed structure, you need to adapt the “init\_mongo.conf” and “mongo.conf” configuration files.

Native working with the mongo shell

To start the mongo shell to connect as administrator use:  
> mongo -u "databaseAdmin" -p "abc123" --authenticationDatabase "admin"

To start the mongo shell to connect as application user use:  
> mongo -u voting\_app -p 12345678 --authenticationDatabase application\_users

Installation of production environment

To fulfill the minimum requirements of MongoDB specification for a production system, the application will run on a replication set consisting of three MongoDB instances. To keep the installation as simple as easy, default scripts for the basic setup are provided.

Note: The scripts are examples to setup a replication set on a single host. For real production setup follow the instructions on [<https://docs.mongodb.com/v2.6/tutorial/deploy-replica-set-with-auth/>] and install each MongoDB instance on a separate host.

Prepare the file system to host the MongoDB

On the root directory create a directory structure like shown in the picture. Place the initialization scripts from the tools repository in the “prod” directory.

Optional:  
The tools repository provides a default keyfile. When you create a new keyfile for the usage on Windows hosts, use the old “cmd.exe” to call > openssl

MongoDB expects an ASCII or UTF-8 encoded keyfile. While the “cmd.exe” generates this, the new “powershell.exe” of Windows creates UTF-16 encoded files which are not usable by MongoDB.

Setup the database.

1. Start the first MongoDB via command prompt  
   > mongod –config mongo1.conf  
   Check if the start was successful via “log/database/node1.log
2. Open a new command prompt and start the second instance  
   > mongod –config mongo2.conf  
   Check if the start was successful via “log/database/node2.log
3. Open a new command prompt and start the third instance  
   > mongod –config mongo3.conf  
   Check if the start was successful via “log/database/node3.log
4. Open a new command prompt and execute the js file via mongo shell   
   > mongo 01\_init\_replica\_set.js  
   This step takes a while to complete  
   Check if the initialization on each node was successful
5. Open a mongo shell by typing “mongo” and check the status of the replication set  
   > rs.status()  
   Check the host and port number of the PRIMARY node, see image
6. Execute the following js scripts via mongo shell   
   Note: check if you need an additional “--port” or “--host” parameter to execute these scripts on the primary node  
   > mongo 02\_prepare\_database.js  
   > mongo 03\_databasae\_user.js  
   > mongo 04\_init\_database.js
7. Import an initial data set for the first application administrator with mongoimport (optional)  
   > mongoimport -u voting\_app -p 12345678 --authenticationDatabase application\_users -d voting\_poc -c users < users.json

Note:  
If your directory structure is differing from the proposed structure, you need to adapt the “.conf” configuration files.  
When you work with differing hosts or port numbers you need to adapt the “.conf” configuration files and “01\_init\_replica\_set.js”.

MongoDB is now ready to be used in production mode.

Important:  
Names of users and databases are suggestions for the PoC and needed to be adapted according to enterprise standards and compliance during setup of the actual database

Native working with the mongo shell

To start the mongo shell to connect as administrator use:  
> mongo voting\_poc -u "databaseAdmin" -p "abc123" --authenticationDatabase "admin"

To start the mongo shell to connect as application user use:  
> mongo voting\_poc -u voting\_app -p 12345678 --authenticationDatabase application\_users

Script overview:

The scripts for development and production setup both execute the same setup:

1. An administrator is added who has the privileges to manage other users
2. A new database called “application\_users” is created  
   Database “application\_users” is the central store of all technical user accounts which are used by applications to secure the access to the database.
3. A new application user “voting\_app” is added to the database.
4. Index creation  
   The application uses the email address as unique identifier for authentication. Therefore a unique index is created on the field in the according “users” collection.

On production setup an additional cluster administrator is created.