# Sharepoint.lanit-task

Solution contains 5 main parts:

* Configured list schema “Site Columns\Statistics”
* Events Listeners “Site Columns\StatisticsEvents”
* Service “ISAPI\StatisticsServices”
* Webpart “Webparts\StatisticsDetails”
* Webpart “Webparts\StatisticsChart”

Some service modules:

* Bootstrap module
  + Provides styles and scripts for bootstrap library
* StatisticsChartModule module
  + Provides styles and scripts for vuejs chart webpart

# Common

IDE used for development:

* Backend – VisualStudio 2015 v14.0.23
* Frontend – PhpStorm 2017.1.4

All Settings data stores in “App/Settings.cs”

All Classes, Workers and Models stores in “App”

All Exceptions, Errors and Log messages writes into system events with provider “*SharePoint Custom Services*”

[s6]

**Site Columns\Statistics**

List Contains of 3 columns:

* **sValue** [Number] - store number values typed by user in Webpart “Webparts\StatisticsDetails”
* **sDate** [Date and Time] – store date and time values selected by user in
* “Webparts\StatisticsDetails”
* **sUser** [Person or Group] – store user object of author

All list fields are read only and not allowed for editing from default list view.

On “FeatureActivated” event some security actions on list are happens:

* Removing all security groups for created list
* Creating new group “Редакторы статистики” and adding current user in it (user that deploying the solution)
* Assigning new group to list

Event Receiver - sharepoint.lanit\_task.Features.Feature1/Feature1EventReceiver/ FeatureActivated

[s2]

# Site Columns\StatisticsEvents

This event listener contains 1 override method “ItemAdding”. During item adding, *cValue* field value filled by user incrementing by 1 and for *cUser* field value assigning user login name with service prefix. All other fields stay the same.

Event Receiver - sharepoint.lanit\_task.StatisticsEvents/StatisticsEvents/ItemAdding

# ISAPI\StatisticsServices

Made by this instructions - [link](https://social.technet.microsoft.com/wiki/contents/articles/24194.sharepoint-2013-create-a-custom-wcf-rest-service-hosted-in-sharepoint-and-deployed-in-a-wsp.aspx). I’m not sure this is a best practice. Service Host manifest file contains a PublicKeyToken key-value pair and, if you have any errors with it, you need to generate it by yourself. Follow this manual – [link](https://blogs.msmvps.com/windsor/2011/11/04/walkthrough-creating-a-custom-asp-net-asmx-web-service-in-sharepoint-2010/).

Service implements 3 Operation Contracts:

* HelloWorld – for tests
* GetUserStatistics/{id} – returning statistics of user with ID = {id}
* GetCurrentUserStatistics – deprecated. Returning current logged in user statistics and requires user to be authenticated.

All Get Methods are allowed by link having mask [http://site/\_vti\_bin/StatisticsServices.svc/{method}/{params}](http://site/_vti_bin/StatisticsServices.svc/%7bmethod%7d/%7bparams%7d)

[s3]

Attention:

For using services, Anonymous Access must be turned on in SharePoint WebSite.

Run IE as admin, go to SharePoint Administration Center, to *\_admin/WebApplicationList.aspx*, select Application on 80 port and follow instructions on screenshots:

[s8]

Code Behind - sharepoint.lanit\_task.Services/StatisticsServices

# Webpart “Webparts\StatisticsDetails”

[s4]

Visual webpart. It is made like a form with 2 input fields – Value and Date. All fields have hidden Required and CompareByType Validators. Bootstrap was used for faster decoration.

After adding item, page refreshes with status code and message, associated for this code.

Command for creating new list item runs with elevated privileges.

[s5]

Code Behind - sharepoint.lanit\_task.Webparts.StatisticsDetails/ StatisticsDetailsUserControl/btnSave\_Click

# Webpart “Webparts\StatisticsChart”

Visual webpart. Contains page with vuejs application on it, that gets data for current user from StatisticsService and display it as a chart. .ascx page contains link to .css and .js files of frontend project, that deploy to SharePoint as a module. <div id=”app”></div> is an entry point for vuejs component.

TODO: for versioning, need to find a solution to include dynamically generated .css and .js files by their names into .ascx. At current time, you need to refresh page with ctrl + f5 after deploying to download new .css and .js.

[s7]

# Frontend

Statistics chart is based on [vue-chartjs](https://github.com/apertureless/vue-chartjs).

“src/App.vue” is a root component, where my *StatisticsChart* component including with *backendUrl* property (url to */\_\_vti\_bin/StatisticsService.svc/GetUserStatistics*). This property contains a link from where script will receive data by get http request. For http requests [axios](https://github.com/axios/axios) lib were used.

Script starts from *mounted* stage:

* Getting *userId* from \_*spPageContextInfo* object
* Running async function *loadStatistics*
* After *loadStatistics* execution, calling parent vue-chartjs method and passing data for display in it.

Source Code of my custom component - src/components/StatisticsChart.js

# Running Frontend Project

For running Frontend project:

* Install [nodejs](https://nodejs.org/en/)
* Install [yarn](https://yarnpkg.com/lang/en/) or use internal nodejs npm
* Install [python 3](https://www.python.org/downloads/windows/)
* In console, go to frontend/statistics-chart folder
* Run command “*yarn install/npm install*” for downloading dependencies
* Run command “*yarn f/npm run f*” for building module in folder “build”
* Place all content of “build” folder into “VSProject/StatisticsChartModule” (clear it before)
* Reload VS Project

# Some comments:

For faster working, [vue-cli](https://github.com/vuejs/vue-cli) was used. And it was first mistake – some compiled files have symbol “~” in names, that is now allowed by VS and SharePoint, where “~” is a special character. Modification of vue-cli source code was a bad idea and I made a python script called “prepare-module.py”. This script renames all files how I need and generates *Elements.xml* and *SharePointProjectItem.spdata* for simple import into VS Project as a module.

In next time, I will use my custom webpack scripts, like here - [link](https://github.com/DreadfulBot/vuejs.quiz)

To run this script, run command “*yarn m/npm run m*” in cmd, or directly python command “*py -3 prepare-module.py*” after building project.