**Department of Veterans Affairs (VA):**

**Office of Informatics & Analytics**

**Connected Health Office, Innovation Program**

**The Daily Plan®**

**Installation Guide**

****

**September 2016**

**Version 1.0**



**1101 King Street, Alexandria VA 22314**

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 11/16/2015 | 1.0 | Initial Version | Robert Snelling |
| 12/23/2015 | 1.0 | Updates | Robert Snelling |
| 01/19/2016 | 1.0 | Updates | Vasudha K Upadhyaya |
| 02/09/2016 | 1.0 | Updates | Vasudha K Upadhyaya |
| 03/11/2016 | 1.0 | Updates | Robert Snelling |
| 04/15/2016 | 1.0 | Updates | Robert Snelling |
| 05/10/2016 | 1.0 | Updates | Vasudha K Upadhyaya |
| 06/17/2016 | 1.0 | Updates | Vasudha K Upadhyaya |
| 07/18/2016 | 1.0 | Updates | Vasudha K Upadhyaya |
| 08/10/2016 | 1.0 | Updates | Vasudha K Upadhyaya |
| 09/07/2016 | 1.0 | Updates | Vasudha K Upadhyaya |

**Table of Contents**

1. About This Document 3

2. Introduction 3

2.1. Purpose 3

2.2. Scope 3

2.3. Software 3

3. Installation Manual 4

3.1. Pre- requisites 4

3.1.1. Server 4

3.2. Pre-installation Tasks 4

3.3. Installation Overview 4

3.4. TDP Application Server 5

3.4.1. Pre-Install Checks 5

3.4.2. Configure MySQL 5

3.4.3. Install Node.js 7

3.4.4. TDP Application Setup 7

3.4.5. TDP App Manual Startup 9

3.4.6. Process Management using PM2 9

3.4.6.1. Reverse Proxy for SSL (Security) 10

3.4.7. TDP Application Installation in case Github has issues 11

3.4.8. Adding a new Facility 12

3.5. Pilot Location Installation 13

3.5.1. KIDS Install 13

3.5.2. EWD.js Install 13

3.5.3. EWD.js Manual Startup 15

3.5.4. EWD.js Unattended Configuration 15

4. Post-Installation 15

4.1. Technical Tests 15

5. Uninstall 19

6. Contact Information 19

7. Acronyms 19

# About This Document

This document is a “work in progress.” In accordance with industry-standard Agile best practices, this document will be populated with content as the design of the application’s architecture evolves with each sprint. As development continues, any activity that requires a change in the installation procedure for the application and related components will require a subsequent update to this document.

# Introduction

This guide is developed for The Daily Plan® (TDP) project for the Department of Veterans Affairs (VA), VA Center for Innovations (VACI), and the Veterans Health Administration (VHA) Innovation Program.

The Daily Plan® (TDP) originated as a patient-specific plan used each day in the hospital. It is a one or two page printed document reviewed with each patient daily by his/her nurse and other health care providers to ensure an accurate treatment plan. Items in TDP are extracted from the electronic medical record and appear as they do in the electronic medical record, e.g., abbreviations and medical jargon not familiar to the patient. TDP includes several items, such as allergies, medications, scheduled procedures, and diet. As a patient safety strategy, patients need to know what to expect before they can advocate for themselves and identify an unplanned or unexpected event.

## Purpose

The purpose of this installation guide is to provide instructions for use by the technical staff who will install The Daily Plan®.

## Scope

The document is intended for system administrator level technical staff with knowledge and experience in database and application installation and configuration.

## Software

Table 1: Software Components

|  |  |
| --- | --- |
| Technology | Technical Reference Model (TRM) Status |
| Node.js (v4.x) | Approved w/Constraints [1] |
| Angular.js | Approved |
| Enterprise Web Developer (EWD).js | Approved w/Constraints [1] |
| MySQL Database | Approved w/Constraints [1] |
| Nginx | Approved w/Constraints [1] |
| KIDS Package |  |

# Installation Manual

## Pre- requisites

All of the assumptions for the pre-requisites have not been determined. The document will be updated as soon as the information becomes available.

### Server

* Linux Virtual Server (Version TBD)
* MySQL Pre-Installed
* Elevated user with SUDO access

## Pre-installation Tasks

* Access to the internet and internal VA network including the VA Secure Repo
* Any web filters, browser configurations, or network restrictions must be mitigated to facilitate the download of software from the internet and internal VA network.

## Installation Overview

The TDP Application is a centralized server with connections to each of the participating pilot locations. The TDP Application has several process that will run on the server. Below is a table that list each of the process:

| **Process** | **Description** | **Port** |
| --- | --- | --- |
| TDP App | Web Application | 80 |
| TDP EWD Interfaces | REST API for Each Pilot Location (4 initially) | 8082 to 8085 |

The installation will include the setup of the centralized server and at each of the pilot locations .

Pilot Location

EWD (Cache)

TDP

Server

(App & DB)

KIDS PKG

(VistA)

## TDP Application Server

All of the TDP installation steps will be done at in a terminal session. Depending on the authority of the user provided to do the install it may be required to do a “sudo” before each command. The setup of the application server entails the following:

* Pre-install Checks
* Configure MySQL
* Install Node.js
* Install and Configure TDP Application

### Pre-Install Checks

1. Check version of Linux
   1. $ cat /etc/\*-release
2. Login and Verify SUDO Access
   1. $ visudo (Look for 'wheel' to see if current user is included)

### Configure MySQL

1. Verify MySQL is installed

$ mysqladmin -u root -p status

Output if Running:

Enter password:

Uptime: 4 Threads: 1 Questions: 62 Slow queries: 0 Opens: 51 Flush tables: 1 Open tables: 45 Queries per second avg: 15.500

Output if Not Running:

mysqladmin: connect to server at 'localhost' failed

error: 'Can't connect to local MySQL server through socket /var/lib/mysql/mysql.sock' (2)'

Check that mysqld is running and that the socket: '/var/lib/mysql/mysql.sock' exists!

1. Verify MySQL root password is set

$ mysql -u root

mysql> SELECT User, Host, Password FROM mysql.user;

1. If needed, set MySQL root password

$ mysql -u root

mysql> UPDATE mysql.user SET Password = PASSWORD('new\_password') WHERE User = 'root';

mysql> FLUSH PRIVILEGES;

1. Secure the MySQL installation with mysql\_secure\_installation script

$ /usr/bin/mysql\_secure\_installation

- Follow prompts to remove test db and anonymous user

1. Set MySQL to auto start (Instructions for Centos)

Note: Instructions are condensed from <https://www.digitalocean.com/community/tutorials/how-to-configure-a-linux-service-to-start-automatically-after-a-crash-or-reboot-part-1-practical-examples>

To auto start services in Centos or Redhat OS, you can use built-in chkconfig utility. It is located in /sbin directory. If you are a regular user (non-root), then /sbin may not be in your path. Therefore, you may have to use the full path to access the chkconfig utility.

**To auto start a new service:**

1. Find out the name of service’s script from /etc/init.d/ directory e.g. mysqld or httpd
2. Add it to chkconfig

$ sudo /sbin/chkconfig --add mysqld

1. Make sure it is in the chkconfig.

$ sudo /sbin/chkconfig --list mysqld

1. Set it to auto start

$ sudo /sbin/chkconfig mysqld on

**To stop a service from auto starting on boot**

$ sudo /sbin/chkconfig mysqld off

1. Create the TDP database and user

$ mysql -u root -p (enter root pw when prompted)

mysql> create database tdpappdb;

mysql> grant all on tdpappdb.\* to tdpappuser@'localhost' identified by 'XXXXXX';

mysql> flush privileges;

### Install Node.js

Install Node.js via NVM (Node Version Manager)

Note are condensed from https://www.digitalocean.com/community/tutorials/how-to-install-node-js-with-nvm-node-version-manager-on-a-vps

$ curl https://raw.githubusercontent.com/creationix/nvm/v0.11.1/install.sh | bash

$ source ~/.bash\_profile

$ nvm install 4.3.2

$ which node -- This will show info on current installation

$ n=$(which node);n=${n%/bin/node}; chmod -R 755 $n/bin/\*; sudo cp -r $n/ {bin,lib,share}

-- This is supposed to copy the current version of node to a production quality directory (/usr/local/)

### TDP Application Setup

The current plan is to install the TDP Application from the secure GitHub Repository but it is also possible to download a zip of the application and install from that.

**Install Git and Clone the TheDailyPlan Repo**

Run the following commands in order as they appear:

$ sudo yum install git-all

$ git config --global user.name “John Doe"

$ git config --global user.email “john.doe@va.gov”

$ cat ~/.gitconfig

-- Verify Output

[user]

name = John Doe

email = john.doe@va.gov

$ cd /usr/src

$ sudo git clone https://username@github.com/VHAINNOVATIONS/TheDailyPlan

-- Will be prompted for git password

$ sudo git checkout master

Branch master set up to track remote branch dev from origin.

Switched to a new branch 'master'

$ sudo git pull

Password: xxxxxxxxxx

**Optional: Install Python 2.7**

<https://dmngaya.com/2015/10/25/installing-python-2-7-on-centos-6-7/>

**Optional: Install SASS**

<http://sass-lang.com/install>

$ sudo su -c "gem install sass"

**Build the Application**

Run the following commands in order as they appear:

$ cd /usr/src/TheDailyPlan/tdpApp

**.env File**

TDP\_VISTA\_ACCESS\_TYPE=EWD

NODE\_ENV=production

# TDP\_RDK\_URL=

TDP\_EWD\_REST\_ALIAS=Biloxi^Madison

TDP\_EWD\_REST\_PORT=8082^8083

EWD\_WEB\_SERVICE\_NAME=vista

EWD\_WEB\_SERVICE\_MODULE=raptor

EWD\_WEB\_SERVICE\_METHOD=parse

EWD\_SERVER\_NAME=tdp

EWD\_SERVER\_HOST=54.158.47.205

EWD\_SERVER\_PORT=8080

EWD\_SERVER\_SSL=false

EWD\_SERVER\_SECRET\_KEY=TakeARest!

EWD\_SERVER\_ACCESS\_ID=RESTServer

TDP\_DB\_DATABASE=tdpappdb

TDP\_DB\_USER=tdpappuser

TDP\_DB\_PW=XXXXXXXX

TDP\_DB\_HOST=localhost

TDP\_DB\_PORT=3306

TDP\_DB\_DIALECT=mysql

TDP\_DB\_POOL\_MAX=5

TDP\_DB\_MIN=0

TDP\_DB\_IDLE=10000

$ npm install -g grunt-cli

$ npm install -g bower

$ npm install

$ bower install

**Configure the Application**

Create a .env file or copy from the secure VA repository.

$ cd /usr/src/TheDailyPlan/tdpApp

$ sudo touch .env

$ sudo vi .env (VI or another text editor).

Create the file to have the following environment variables but with production values (see box to right). After the .env file is in place copy it to the /usr/src/TheDailyPlan/backend/tdpewdrest.

$ sudo cp .env /usr/src/TheDailyPlan/backend/tdpewdrest/.env

**Initialize the Database**

Now that the environment variables are set, the database can be initialized via the script found at /usr/src/TheDailyPlan/tdpApp/server/db/syncAndLoad.js. This will build the database schema and populate it with default data.

$ cd /usr/src/TheDailyPlan/tdpApp

$ sudo node server/db/syncAndLoad.js

After script is finished loading, do a CTRL+C to terminate.

### TDP App Manual Startup

To run the application manually, each of the following sections will require their own terminal session.

**Run the Web Application**

To run the application do the following command. This will compile and start the application on port 9000 by default.

$ sudo grunt serve

**Run TDP EWD Interface for Pilot Locations**

Each of the Pilot Locations will need a TDP EWD Interface running. The start command requires a port and the ip address of the Pilot EWD Server.

$ sudo node app.js 8082 XXX.XXX.XXX.XXX (Pilot Location 1)

$ sudo node app.js 8083 XXX.XXX.XXX.XXX (Pilot Location 2)

$ sudo node app.js 8084 XXX.XXX.XXX.XXX (Pilot Location 3)

$ sudo node app.js 8085 XXX.XXX.XXX.XXX (Pilot Location 4)

### Process Management using PM2

**Online Documentation**: <https://github.com/Unitech/pm2>

PM2 is a production ready process manager for Node.js applications with a built-in load balancer. It allows you to keep applications alive forever, to reload them without downtime and to facilitate common system admin tasks.

Follow the steps to install PM2 and start the application:

$npm install pm2 –g

$cd /usr/src/TheDailyPlan/tdpApp/dist

$ pm2 start app

$ cd /usr/src/TheDailyPlan/backend/tdpewdrest

Create a script file for each of the corresponding the eWD sites

$ vi Pilot1.sh

Insert sudo node app.js 8082 XXX.XXX.XXX.XXX (Pilot Location 1)

Save Pilot1.sh

$vi Pilot2.sh

Insert sudo node app.js 8083 XXX.XXX.XXX.XXX (Pilot Location 2)

Save Pilot2.sh

$vi Pilot3.sh

Insert sudo node app.js 8083 XXX.XXX.XXX.XXX (Pilot Location 3)

Save Pilot3.sh

$vi Pilot4.sh

Insert sudo node app.js 8083 XXX.XXX.XXX.XXX (Pilot Location 4)

Save Pilot4.sh

$pm2 start Pilot1.sh

$pm2 start Pilot2.sh

$pm2 start Pilot3.sh

$pm2 start Pilot4.sh

**Firewall Settings**

**Iptables** is a user-space application program that allows a system administrator to configure the tables provided by the Linux kernel firewall and the chains and rules it stores.

### Reverse Proxy for SSL (Security)

**Online Documentation:** <https://www.nginx.com/resources/admin-guide/reverse-proxy/>

**NGINX** is a free, open-source, high-performance HTTP server and reverse proxy. NGINX is known for its high performance, stability, rich feature set, simple configuration, and low resource consumption.

Please follow the URL to install Nginx on the Server : <https://www.nginx.com/resources/wiki/start/topics/tutorials/install/>

Follow the below instructions to configure Nginx:

$cd /etc/nginx/conf.d/

$ vi virtual.conf

Update the virtual.conf accordingly

|  |
| --- |
| server {  listen 80;  server\_name $ServerName;  access\_log /var/log/nginx/$LogName;  location / {  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header HOST $http\_host;  proxy\_set\_header X-NginX-Proxy true;  proxy\_pass http://127.0.0.1:9000;  proxy\_redirect off;  # root html;  # index index.html index.htm;  }  } |

To start NGNIX use the following command:

/etc/init.d/nginx start

Follow the steps defined below to change the set up secure Server.

Substitute ***server*** with the Actual Server that is being configured

1. Generate the Key and CSR using the command

openssl req -new -newkey rsa:2048 -nodes -keyout ***server***.key -out ***server***.csr

1. Submit the ***server***.csr and request VA for server certificates and wait to receive the ***server***.crt.
2. Get the cert add the following to the ***server***.crt

-----BEGIN CERTIFICATE-----

Payload of ***server***.crt

-----END CERTIFICATE-----

1. Create the following directory under /etc/nginx

sudo mkdir /etc/nginx/ssl.key

sudo mkdir /etc/nginx/ssl.crt

sudo mkdir /etc/nginx/ca-bundle

1. Copy the ***server***.key under /etc/nginx/ssl.key
2. Copy the ***server***.crt under /etc/nginx/ssl.crt
3. Download the Internal Certs(InternalNetCA1.cer, InternalSubCA1.cer,InternalSubCA2.cer,VAInternalRoot.cer and copy under /etc/nginx/ca-bundle
4. Convert them to .pem files and bundle them together:

openssl x509 -inform PKCS7 -outform PEM -in VAInternalRoot.cer -out VAInternalRoot.pem

openssl x509 -inform DER -outform PEM -in InternalSubCA1.cer -out InternalSubCA1.pem

openssl x509 -inform DER -outform PEM -in InternalSubCA2.cer -out InternalSubCA2.pem

cat InternalSubCA1.pem InternalSubCA2.pem VAInternalRoot.pem > VABundle.pem

1. Update the virtual.conf file to add the following lines

cd /etc/nginx/conf.d

vi virtual.conf

# listen 80;

listen 443 ssl;

server\_name ***server***

ssl\_certificate /etc/nginx/ssl.crt/ ***server***.crt;

ssl\_certificate\_key /etc/nginx/ssl.key/ ***server***.key;

ssl\_trusted\_certificate /etc/nginx/ca-bundle/VABundle.pem;

ssl\_ciphers HIGH:!aNULL:!MD5;

1. Save the file and restart nginx using the command

$/etc/init.d/nginx restart

1. Start using the secure(https) URL to access the application.

### TDP Application Installation in case Github has issues

If the TDP Application server has issues with Github install the application on the Dev Server and then upload the files to the IT Pilot Server

Steps to retrieve the files from the Dev/Test Server:

$cd /usr/src/TheDailyPlan/tdpApp/dist/

Backup the .env file

$ pm2 stop app

$ sudo git checkout master

$ cd /usr/src/TheDailyPlan

$ sudo git pull

Password: Enter Github Password

The Update files are copied.

cd tdpApp/

Copy the backed up .env file under /usr/src/TheDailyPlan/tdpApp/dist/ and /usr/src/TheDailyPlan/tdpApp/dist/server

$ sudo grunt build

$ sudo node dist/server/db/syncAndLoad.js

$ cd /usr/src/TheDailyPlan/tdpApp/dist/public/common/assets/images

$ sudo cp TDP-logo-lg.png TDP-Logo-tr-lg.png

$ pm2 start app

Make sure the Application is up and running.

Copy the dist folder to the IT Pilot Server and follow the step below:

$ pm2 stop app

$cd /usr/src/TheDailyPlan/tdpApp/dist/

Backup the .env file

$ cd /usr/src/TheDailyPlan/tdpApp/

$ rm –rf dist/

Get the dist folder from the Dev Server

$sudo cp –r dist/ /usr/src/TheDailyPlan/tdpApp/

Change the owner to the Current Logged in Owner using the .command :

$sudochown –R $CurrentOwner:root dist

Copy the backed up .env file under /usr/src/TheDailyPlan/tdpApp/dist/ and /usr/src/TheDailyPlan/tdpApp/dist/server

$ sudo node dist/server/db/syncAndLoad.js

$ pm2 start app

### Adding a new Facility

1. To add a new Facility, a record needs to be inserted in the database. Follow the steps below:

cd /usr/src/TheDaily/Plan/tdpApp/server/db

vi addFacility.js

Go to line 779 and change the facility name as needed ( example ‘NewFacility’)

Save the file

cd tdpApp/server

sudo node db/addFacility.js

1. Update the .env file.

cd /usr/src/TheDaily/Plan/tdpApp/

vi .env

Add the name of the facility and an unused local port to the line listed below:

Before

TDP\_EWD\_REST\_ALIAS=Biloxi^Madison

TDP\_EWD\_REST\_PORT=8082^8083

After

TDP\_EWD\_REST\_ALIAS=Biloxi^Madison^NewFacility

TDP\_EWD\_REST\_PORT=8082^8083^8084

1. Run a new rest server for NewFacility as described in Section 3.5
2. Restart the TDP Server

pm2 restart app

## Pilot Location Installation

### KIDS Install

Each Pilot Location will need to apply the TDP KIDS Package to their local VistA System. The TDP Team will rely on local admins to apply the KIDS Package based on their local guidelines.

### EWD.js Install

The Pilot Location EWD.js install will need to occur on the same server that Caché is running on. The following is for linux installation for EwD version 0.103.0 with which TDP is tested. Windows installations are similar

1. Create a 'ewdjs' directory where ewdjs installation will be placed. In the following it will be assumed that this directory is in your home directory (~/ewdjs)
2. cd ~ewdjs
3. npm install ewdjs@0.103.0
4. When you are asked for installation directory accept the default.
5. When you are asked to install additional resources enter 'Y'.
6. Locate 'cache0100.node' file from your Cache installation. This is expected to be in you Cache/bin directory. If you cannot find it contact to your Cache representative.
7. cp <cache0100.node> ~/ewdjs/node\_modules/cache.node (it is important to name the file as cache.node in the node\_modules directory). EWD.js installation has been completed.

Please refer to the attached document for further information.



Upon completion of the EWD.js install, copy all of the TDP EWD Modules in to the ~ewdjs/node\_modules/ folder. These files reside in <https://github.com/VHAINNOVATIONS/TheDailyPlan/tree/master/backend/ewdmodules> in The Daily Plan repository. Below is a list of files as of the writing of this document:

VistALib.js

tdp.js

vistaAllergiesLib.js

vistaChemHemLib.js

vistaHealthFactors.js

vistaMedsLib.js

vistaOrders.js

vistaPatientLib.js

vistaPatientSearch.js

vistaPostingsLib.js

vistaTiuLib.js

The completion of the installation will result in a default configuration of:

http://X.X.X.X:8080/ewd/ewdMonitor/index.html

password: keepThisSecret!

var params = {

  poolSize: 2,

  httpPort: 8080,

  traceLevel: 3,

  database: {

    type: 'cache',

    path:"/srv/vista/mgr",

    username: "\_SYSTEM",

    password: "SYS",

    namespace: "USER"

  },

  management: {

    password: keepThisSecret!

The final configuration of the EWD Installation will be to modify the configuration in **ewdStart-cache-linux.js** which is located in the main EWD folder (~ewdjs/)

* database.namespace: This should set up to point to the Cache namespace that is used by VistA instance.
* management.password: This will be shared by the The Daily Plan client.
* Change other fields as appropriate.

### EWD.js Manual Startup

To run the application manually, each of the following sections will require their own terminal session.

**Run the EWD.js Application**

To run the application do the following command. This will start the application and listen for connections.

$ sudo ~/ewdjs/ewdStart-cache-linux.js

### EWD.js Unattended Configuration

The production configuration is still being formulated but the following components look to be used in that process. This will automate the server to run the above processes without human intervention and will provide for load balancing, logging, and security management.

**Process Management**

PM2 is a production process manager for Node.js applications with a built-in load balancer. It allows you to keep applications alive forever, to reload them without downtime and to facilitate common system admin tasks.

# Post-Installation

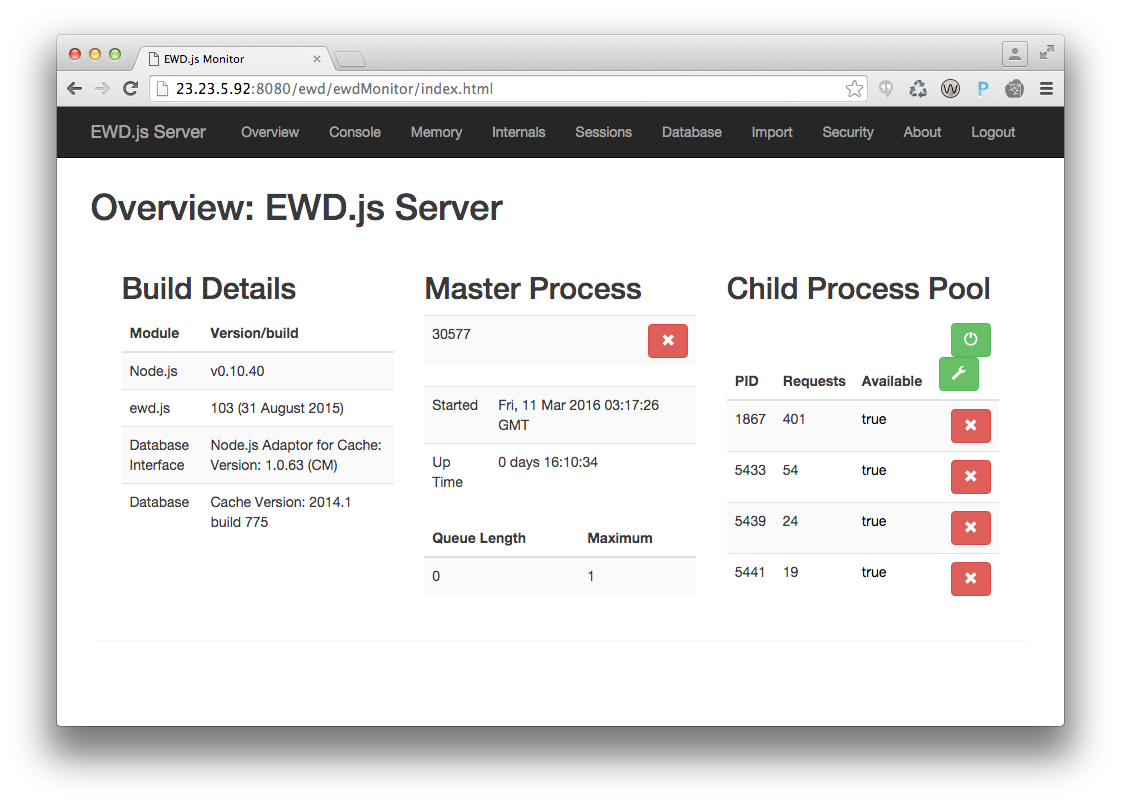
## Technical Tests

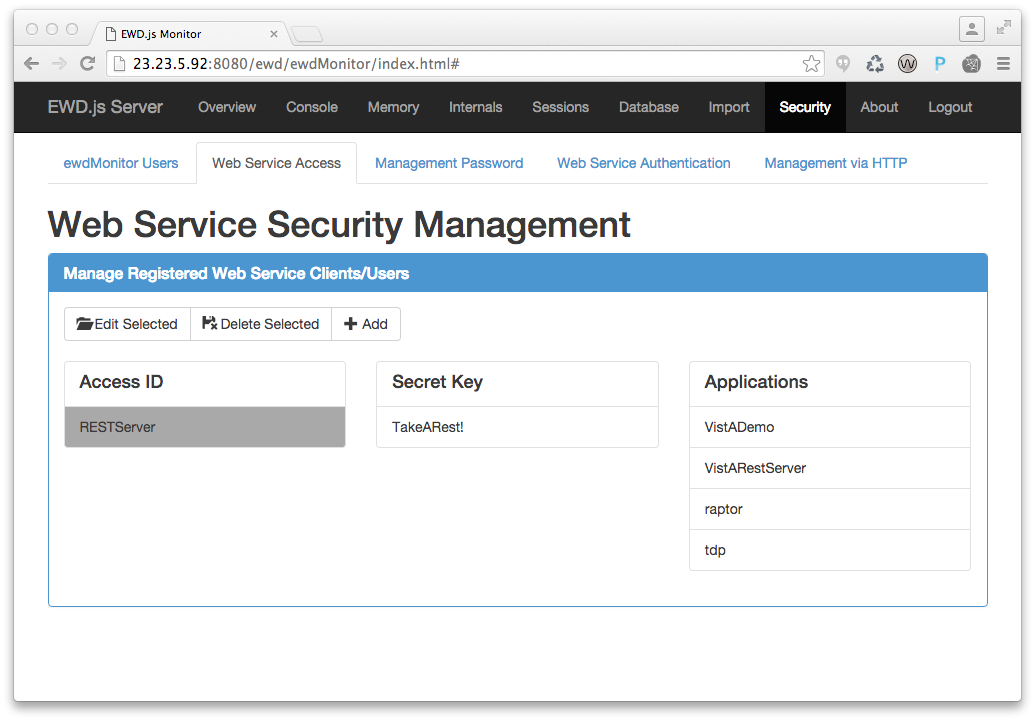
Upon complete of the installation and configuration of the database and application the System Administrator can perform the following tests.

After installation is complete you can check if EwD server properly started by running the EwD monitor

http://$ server ip:8080/ewd/ewdMonitor/index.html

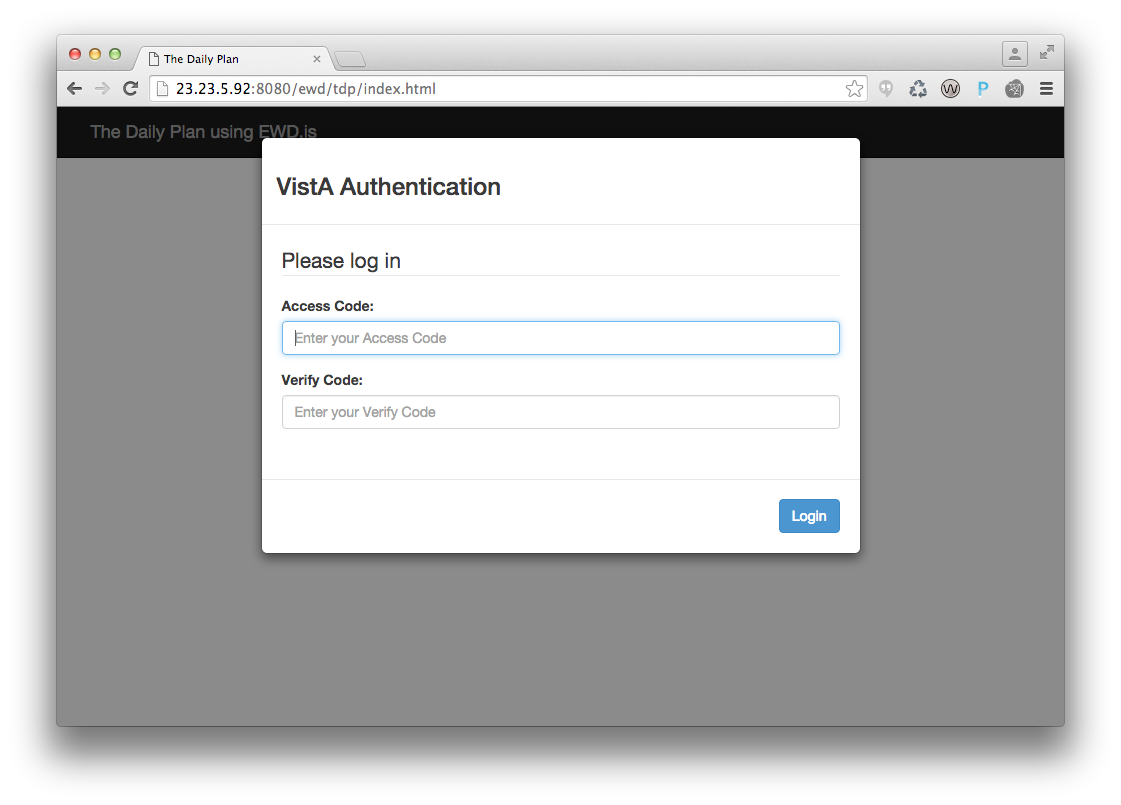
ewdMonitor requires the management.password that is specified in file ewdStart-cache-linux.js. Once you login to ewdMonitor.js and Overview page is shown

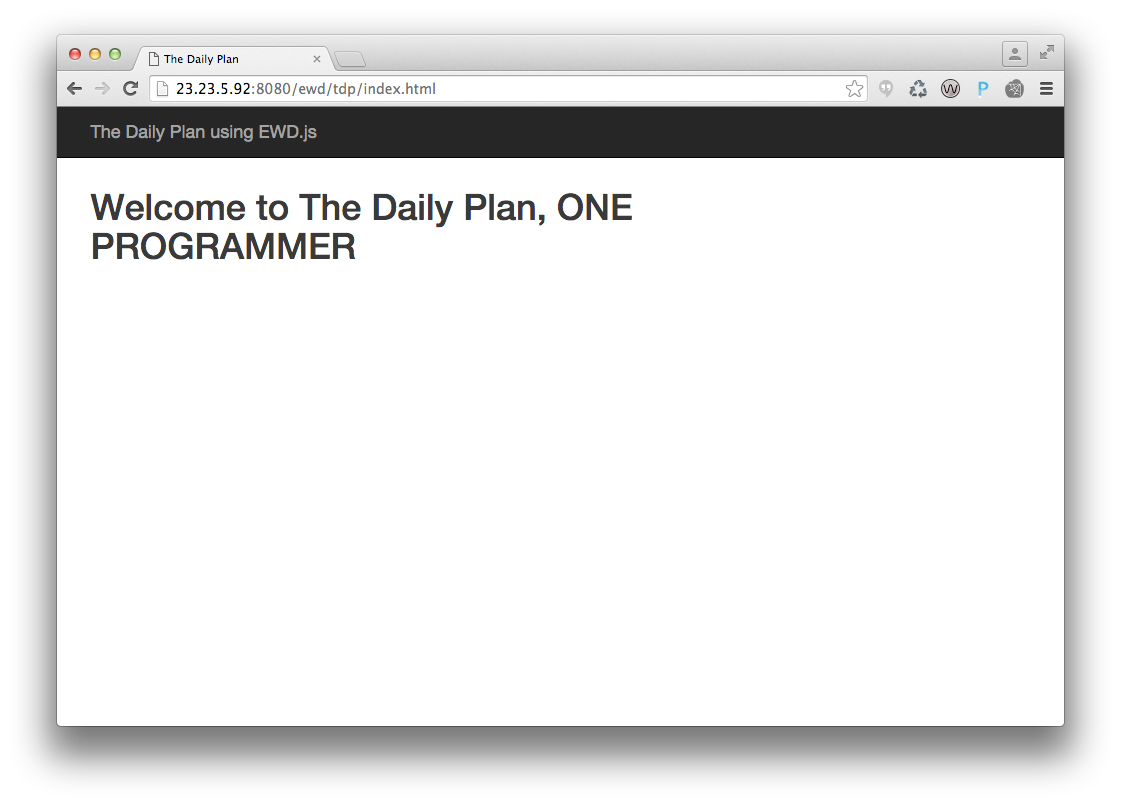


TDP application needs permissions to use EwD based rest services. After installation is complete you need to give permissions to TDP application to use the rest services. This is done in Security/Web Service. Make sure there is a service named RESTServer and Applications include tdp and VistARestServer. SecretKey here is used by tdp rest servers.

Once you complete above steps you can check if the tdp specific components are installed by running the tdp tester

http://23.23.5.92:8080/ewd/tdp/index.html

In this application you can login as any user that has access to the VistA instance. This application simply displays the name of the user. This application uses both EwD components of the TDP and VistA components of TDP and serves as installation test for these components.



# Uninstall

The process of uninstallation represents a complete removal of application and installation files. The administrator is recommended to stop the application and database respectfully. Once the application has been stopped the components and software can be uninstalled by deleting each of the folder that was created in the installation process.

# Contact Information

Table 2: Contacts

|  |  |
| --- | --- |
| Name | Email |
| Vasudha Upadhyaya | vasudha.k.upadhyaya@us.pwc.com |
| Afsin Ustundag | Afsin.ustundag@us.pwc.com |
|  |  |

# Acronyms

Table 3: Acronyms and Definitions

| Term | Definition |
| --- | --- |
| ISO | Information Security Officer |
| TRM | Technical Reference Model |
| VA | Department of Veterans Affairs |
| URL | Uniform Resource Locator |
| VHA | Veterans Health Administration |