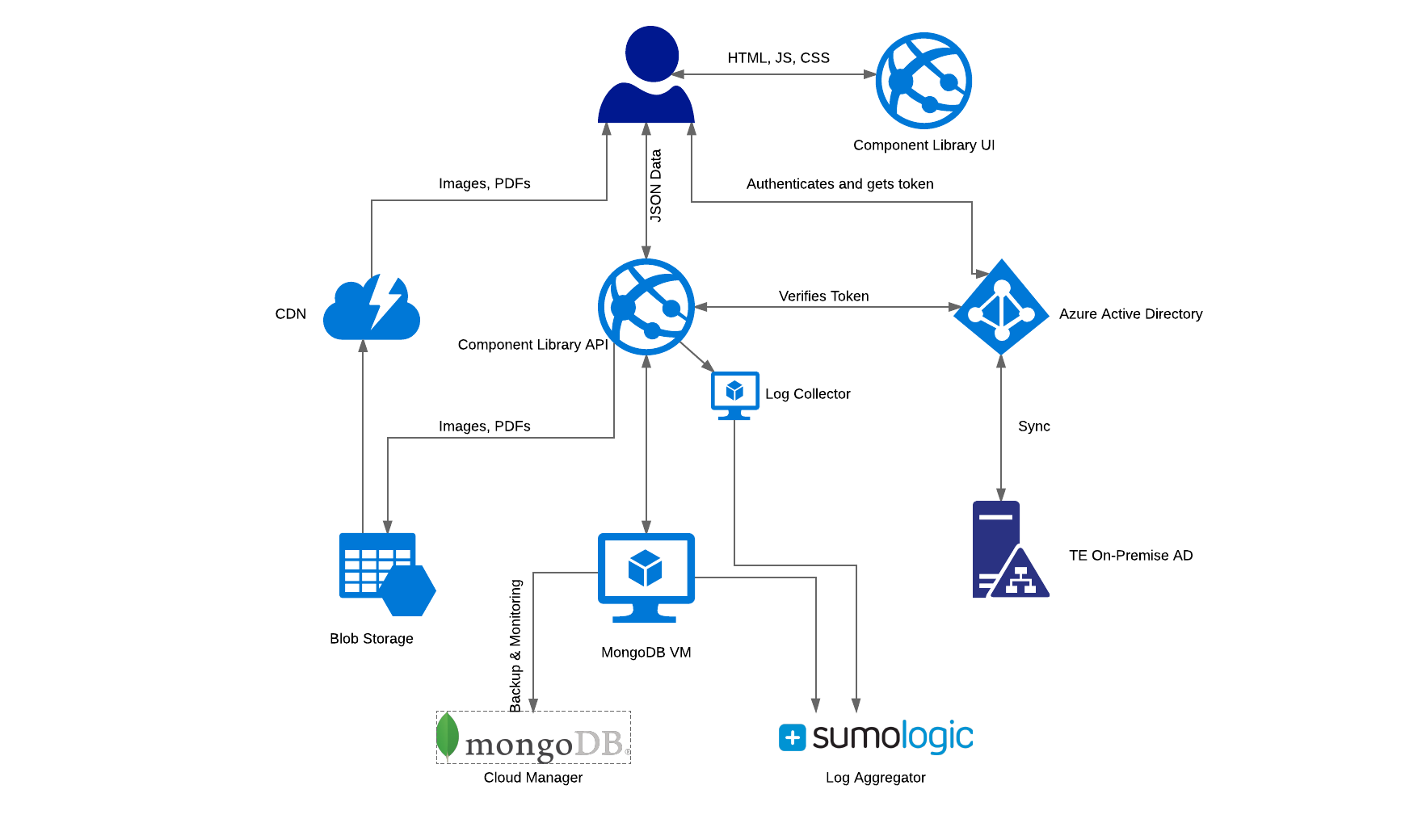
Component Library



## API

URL: <https://clapi.total-environment.com>

Source Code: <https://github.com/TotalEnvironmentProj/ComponentLibrary>

Component Library API is an ASP.NET MVC Web API application written in C# 7. Visual Studio 2017

### Dev Setup

#### Visual Studio 2017

#### Azure SDK

#### .NET Framework 4.6.2

#### MongoDB Community Edition

After installing MongoDB, follow this guide to set it up properly: <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/#configure-a-windows-service-for-mongodb-community-edition>

#### MongoDB Client (Optional)

### Libraries/Frameworks

#### .NET Framework

.NET Framework 4.6.2 is used by Component Library along with C# version 7

#### ASP.NET Web API

ASP.NET Web API 2 is used as the primary web framework

#### Elmah

<https://elmah.github.io/> is used for capturing all the errors.

#### Log4net

Logging library

#### MongoDB C# Driver

Library for connecting to and querying MongoDB.

#### SimpleInjector

SimpleInjector is a Dependency Injection framework

#### XUnit

Testing Framework

#### FluentAssertions

Assertions

#### CloudServiceFramework

Cloud Service Framework is an abstraction over Azure Blob Storage. It’s an internal library that is hosted on our Nuget Server.

#### NotificationEngine

An abstraction over Sendgrid to send emails. It’s an internal library that is hosted on our Nuget Server.

### Build/Deployment

#### Psake

<https://github.com/psake/psake>

run-build.ps1 is the build script that launches build.ps1 that:

1. Compiles the solution using msbuild
2. Runs all the tests
3. Generates Nuget package Component Library

#### Teamcity

<http://tebuild.southeastasia.cloudapp.azure.com/>

TeamCity builds Component Library. It is configured to listen to commits on Github and launches the build process when a new commit is pushed. The build process is described below:

1. Install Nuget Packages
2. Invoke run-build.ps1 to build the project and generate Nuget package
3. Push the Nuget package to Octopus (using an API key)
4. Create a release in Octopus

#### Octopus

<http://tedeploy.southeastasia.cloudapp.azure.com/>

Octopus is the deployment tool. It has one step for each environment to deploy to it. It replaces variables in Web.config with the configured variables while deploying.

#### Nuget Server

<http://twtenugetserver.azurewebsites.net/>

Hosts CloudServiceFramework and NotificationEngine, the internally developed libraries.

### Integrations

#### Azure Active Directory

#### Fugue

#### SAP

### Infrastructure

#### Azure App Service

Component Library API and Component Library UI are both App Services in Azure. Requests from Component Library UI (from the browser) go directly to the API, so, CORS needs to be configured. Component Library UI URL should be listed as a valid host in Component Library API list of CORS (<https://docs.microsoft.com/en-us/rest/api/storageservices/cross-origin-resource-sharing--cors--support-for-the-azure-storage-services>)

#### Azure Blob Storage

Component Library uses Azure blob storage for storing images/PDFs etc. They are stored in a container called static-files.

#### CDN

Files stored in Azure Blob storage are accessed via a CDN

#### MongoDB

MongoDB (version 3.4) is installed on a Linux Virtual Machine. On production, MongoDB is within a virtual network of which Component Library API is also a part.

#### MongoDB Cloud Manager

MongoDB cloud manager is used to manage the MongoDB VM. We first install the MongoDB Automation Agent on the VM through which we install MongoDB itself, the Monitoring agent and Backup agent. Backups are configured only on Production. MongoDB Cloud Manager provides a way to restore any backup with the click of a button.

#### Sumologic

Sumologic is a log aggregator. It provides a central place where all the logs are collected and queried over. Sumologic needs a collector process installed on the machine the logs are generated on. But, since App Service doesn’t allow us to install a process, we need to create a collector VM that has the agent instead. It picks up logs from Blob storage to which the App Service can log.

#### SendGrid

SendGrid is mail delivery agent. It gives a HTTP API to send emails. It also has a template editor. Component Library defines a template called SAP Sync failure which is used to notify SAP support when SAP Sync fails. NotificationEngine is the library that abstracts away the SendGrid away from Component Library.

## UI

URL: [https://cl.total-environment.com](https://clapi.total-environment.com)

Source Code: <https://github.com/TotalEnvironmentProj/ComponentLibrary>

### Libraries/Frameworks:

#### React

#### Redux

#### React-router

#### Axios

#### Moment

#### Babel/Webpack

#### karma/mocha/chai/enzyme

#### Dev Setup

#### Node.js

#### Yarn

After installing yarn, run `yarn install` in the project directory to install all the packages. `yarn start` starts the development server. `yarn test` runs all the unit tests in watch mode.

### Build/Deployment

Webpack is used to build all the Javascript, CSS into one single file. The generated file, bundle.js, along with index-server.html, package-server.json is packaged using the octopack package (see the publish.js file in the source code for details) and pushed to Octopus.

Octopus deploys this package to the App Service and replaces the variables in index.html as part of the process.

### Infrastructure

Express is used as the server for running the web server on production. As such, while creating the App Service, it needs to be created as an Node.js Empty Web App. After creating, through the console on Azure Portal, run the command `npm install express` to install express. Then, deploy the code using Octopus (or trigger a build through Teamcity).