



# Agents of S.W.E.

A SOFTWARE COMPANY

Agents of S.W.E. - Progetto "G&B"

## Manuale dello Sviluppatore

<b>Versione</b>	0.0.3
<b>Approvazione</b>	??
<b>Redazione</b>	Marco Chilese Luca Violato
<b>Verifica</b>	??
<b>Stato</b>	Work in Progress
<b>Uso</b>	Esterno
<b>Destinato a</b>	Agents of S.W.E. Prof. Tullio Vardanega Prof. Riccardo Cardin Zucchetti S.p.A.

agentsofswe@gmail.com

## Registro delle Modifiche

Versione	Data	Ruolo	Autore	Descrizione
0.0.3	2019-04-09	Progettista	Marco Chilese	Stesura di §??, §??, §2
0.0.2	2019-04-06	Analista	Luca Violato	Aggiunte sezioni §2 e §6
0.0.1	2019-04-04	Progettista	Marco Chilese	Strutturazione del Documento. Prima stesura di §4.1, §7.1, §5.1, §5.2, §7.2, §7.3, §3.3

**Tabella 1:** Registro delle Modifiche

## Indice

<b>1</b>	<b>Introduzione</b>	<b>4</b>
1.1	Scopo del Documento . . . . .	4
1.2	Scopo del Prodotto . . . . .	4
1.3	Riferimenti . . . . .	4
<b>2</b>	<b>Principali Tecnologie Utilizzate</b>	<b>6</b>
2.1	Scopo del capitolo . . . . .	6
2.2	Tecnologie Lato Client . . . . .	6
2.3	Tecnologie Lato Server . . . . .	6
2.4	Tecnologie per il Testing . . . . .	6
<b>3</b>	<b>Installazione</b>	<b>7</b>
3.1	Requisiti . . . . .	7
3.2	Impostazione del Prodotto . . . . .	7
3.3	Esecuzione . . . . .	7
<b>4</b>	<b>Impostare l'Ambiente di Lavoro</b>	<b>8</b>
4.1	Scopo del Capitolo . . . . .	8
<b>5</b>	<b>Requisiti</b>	<b>8</b>
5.1	WebStorm . . . . .	8
5.2	ESLint . . . . .	8
<b>6</b>	<b>Architettura</b>	<b>9</b>
6.1	Frontend . . . . .	9
6.2	Backend . . . . .	9
<b>7</b>	<b>Test</b>	<b>10</b>
7.1	Scopo del Capitolo . . . . .	10
7.2	Test sul Codice JavaScript . . . . .	10
7.3	Code Coverage . . . . .	10
<b>8</b>	<b>Licenza</b>	<b>11</b>

## Elenco delle tabelle

1	Registro delle Modifiche . . . . .	1
---	------------------------------------	---



## Elenco delle figure

# 1 Introduzione

## 1.1 Scopo del Documento

Lo scopo del documento è quello di fornire tutte le informazioni necessarie agli sviluppatori che intenderanno estendere o migliorare il plug-in G&B.

Verranno fornite informazioni relative anche ad un possibile ambiente di sviluppo il più completo possibile da cui sarà possibile partire. In particolare sarà illustrato l'ambiente di sviluppo utilizzato dal team **Agents of S.W.E.** per lo sviluppo del prodotto in oggetto.

La seguente guida può essere utilizzata indifferente sia da utenti Microsoft Windows, Linux e Apple MacOS.

## 1.2 Scopo del Prodotto

Lo scopo del prodotto è la creazione di un plug-in per la piattaforma open source<sub>G</sub> di visualizzazione e gestione dati, denominata Grafana<sub>G</sub>, con l'obiettivo di creare un sistema di alert dinamico per monitorare la "liveliness"<sub>G</sub> del sistema a supporto dei processi DevOps<sub>G</sub> e per consigliare interventi nel sistema di produzione del software. In particolare, il plug-in<sub>G</sub> utilizzerà dati in input forniti ad intervalli regolari o con continuità, ad una rete bayesiana<sub>G</sub> per stimare la probabilità di alcuni eventi, segnalandone quindi il rischio in modo dinamico, prevenendo situazioni di stallo.

## 1.3 Riferimenti

### 1.3.1 Referimenti per l'Installazione

- <https://www.npmjs.com/>.

### 1.3.2 Referimenti Legali

- <https://grafana.com/docs/contribute/cla/>.

### 1.3.3 Referimenti Informativi

- <https://grafana.com/>;
- <https://www.influxdata.com/time-series-platform/telegraf/>;
- <https://www.influxdata.com/time-series-platform/influxdb/>;
- <https://jestjs.io/>;



- <https://www.jetbrains.com/webstorm/>;
- <https://webpack.js.org/>.
- <https://angularjs.org/>;
- <http://codecov.io>;
- <https://jquery.com/>.

## 2 Principali Tecnologie Utilizzate

### 2.1 Scopo del capitolo

In questo capitolo verranno esposte e contestualizzate le tecnologie impiegate all'interno del progetto, in modo tale da offrire agli sviluppatori un quadro più completo del progetto.

### 2.2 Tecnologie Lato Client

Lato client vengono adottate le seguenti tecnologie:

- EcmaScript 6: principale linguaggio con cui il plug-in è strutturato, in particolare modo viene utilizzato per lo sviluppo del pannello;
- AngularJS: framework adottato da Grafana per l'interazione con l'utente;
- HTML & CSS: rispettivamente nelle versioni 5 e 3, utilizzati per modellare l'interfaccia del pannello.

### 2.3 Tecnologie Lato Server

Lato server vengono adottate le seguenti tecnologie:

- NodeJS: tecnologia utilizzata per implementare il server.

### 2.4 Tecnologie per il Testing

La tecnologia utilizzata per testare il codice è *Jest*, framework di test per codice JavaScript.



## 3 Installazione

### 3.1 Requisiti

### 3.2 Impostazione del Prodotto

### 3.3 Esecuzione

Per poter utilizzare il prodotto all'interno di Grafana, e poiché quest'ultimo sia riconosciuto come plug-in, è necessario per prima cosa eseguire la build del prodotto, che avviene attraverso l'intervento del module bundler *WebPack* in seguito all'esecuzione di un comando:

```
npm run build.
```



## 4 Impostare l'Ambiente di Lavoro

### 4.1 Scopo del Capitolo

In questa sezione viene riportata una guida per la corretta configurazione dell'ambiente di sviluppo, in modo che sia la stessa utilizzata dal gruppo **Agents of S.W.E.**.

Per contribuire al progetto non è strettamente necessario seguire questa sezione, tuttavia è consigliato al fine di ottenere un ambiente di lavoro pronto e correttamente impostato per lo sviluppo.

## 5 Requisiti

### 5.1 WebStorm

*WebStorm* è l'ambiente di sviluppo integrato (IDE) di JetBrains utilizzato dal team per lo sviluppo del progetto. Esso può essere ottenuto mediante download dal sito ufficiale nella formula di prova gratuita se non si dispone di licenza, che può essere ottenuta attraverso l'indirizzo email universitario.

Tale software è disponibile per i sistemi operativi Microsoft Windows, Linux e Apple MacOS.

Per ulteriori informazioni si rimanda al sito ufficiale.

### 5.2 ESLint

*ESLint* verrà installato automaticamente attraverso il comando `npm install`.

Per abilitarlo all'interno di WebStorm è necessario lanciare l'IDE e recarsi in: File > Settings > ESLint e scegliere "Enable". All'interno del campo "Node Interpreter" è necessario inserire il percorso alla directory in cui si trovano i file eseguibili di Node. Se non rilevato in automatico, specificare la posizione del file di configurazione ".eslintrc" all'interno della directory del progetto.



## 6 Architettura

### 6.1 Frontend

### 6.2 Backend

## 7 Test

### 7.1 Scopo del Capitolo

Questo capitolo ha lo scopo di indicare agli sviluppatori come controllare in modo automatico il proprio codice e la sintassi.

### 7.2 Test sul Codice JavaScript

Per eseguire i test sul codice è necessario eseguire i seguenti comandi:

```
npm run test  
o  
jest.
```

Per verificare che il codice sia coerente con le linee guida adottate è necessario eseguire:

```
npm run eslint.
```

Per correggere in modo automatico alcuni dei potenziali problemi, eseguire:

```
npm run eslint-fix.
```

Questi script eseguono un controllo del codice all'interno di `./src`, la directory dov'è contenuto il codice.

Se si desidera eseguire *ESLint* su di un'unico file, si rimanda il lettore alla documentazione ufficiale.

### 7.3 Code Coverage

Per eseguire il controllo di coperta del codice JavaScript, è necessario eseguire il seguente comando:

```
npm run codecov.
```

Nel caso tale comanda fallisca la principale motivazione è data dall'assenza di report su cui generare il code coverage. Per rimediare a ciò basterà eseguire il primo comando nella sezione §7.2 il quale provvederà a generare i report necessari.



## 8 Licenza