

---

# **HIJSON**

# **Web Framework**

## **Requisiti ed Installazione**

---

Marco Sportillo  
Marco Virgadamo

---

# Requisiti

---

- 1) Installare node.js (<https://nodejs.org/>)
  - 2) Scaricare il Web Framework (WF)  
(<https://github.com/cvdlab/HIJSON>)
-

# Istruzioni per l'avvio

---

- 1) Spostarsi nella root del WF;
  - 2) Eseguire il comando “npm install” (solo la prima volta) per installare le dipendenze;
  - 3) Lanciare il WF con il comando “npm run develop”;
  - 4) Il WF e' raggiungibile all'indirizzo <http://localhost:3000> .
-

---

# HIJSON

# Web Framework

## Introduzione

---

Marco Sportillo

Marco Virgadamo

---

# Introduzione

---

## **Hierarchical Indoor JSON**

**E' un nuovo formato di interscambio per la rappresentazione di ambienti indoor, con supporto all'interattività in contesti di Internet of Things.**

---

# Caratteristiche

---

- **Struttura gerarchica**
  - **Coordinate metriche locali**
  - **Estensione di classi semantiche**
-

# I file HIJSON

---

**Sono tutti nella cartella json\_input e sono (di solito) tre files:**

- config.json
  - architecture.json
  - furniture.json
-

**I file HIJSON**

---

**DEMO**

---



---

# HIJSON

# Web Framework

**Implementazione nuove features**

---

Marco Sportillo  
Marco Virgadamo

---

# Implementazione nuova classe

---

**Sono tutte nella cartella:  
/c3d/features**

---

# Implementazione nuova feature

---

1. copiare il file TemplateFeature.js;
  2. find & replace di ExampleName con il nuovo nome da assegnare all'oggetto, rinominando anche il file. **Usare la lettera maiuscola.**
  3. implementare il metodo get3DModel()
  4. in c3d/modules/featureFactory.js, aggiungere l'entry nell'oggetto *featureclasses*.
-

# Implementazione nuova feature

---

5. in `c3d/modules/renderer2D.js` modificare l'array `orderClasses` (tra le prime righe del file) per aggiungere il nome della classe appena creata.

L'ordine di apparizione nell'array determina la sovrapposizione degli strati nella mappa 2D da quello inferiore a quello superiore.

---

# Implementazione nuova feature

---

E' necessario riscrivere il metodo  
*get3DModel()*.

All'interno della funzione viene  
modellato l'oggetto che sara'  
renderizzato nel modello 3D.

---

# Implementazione nuova feature

---

DEMO

---

# Modellazione nuova feature

---

All'interno della cartella featureEditor trovare un piccolo editor.

Scrivete il vostro metodo nel file *createObject.js*

Potete visualizzarlo da index.html

---

# Modellazione nuova feature

---

Per convenzione, chiamiamo l'oggetto `object`.

Una volta terminato, scommentate le righe alla fine del metodo e cancellate il vecchio *return*.

Copy&Paste nella feature di C3D.

---



# Modellazione nuova feature

---

DEMO

---

# Contatti

---

Marco Sportillo: [sportillo@dia.uniroma3.it](mailto:sportillo@dia.uniroma3.it)

Marco Virgadamo: [virgadamo@dia.uniroma3.it](mailto:virgadamo@dia.uniroma3.it)

---