

# Manuale di Installazione

MyStandard

## SOMMARIO

1	RIFERIMENTI	3
2	GLOSSARIO	3
3	ACRONIMI	3
4	CONTESTO	4
5	OBIETTIVI	4
6	PREREQUISITI INFRASTRUTTURALI	4
7	PREREQUISITI SERVIZI MYPLACE E VERTICALI MYP	34
8	INSTALLAZIONE	5
8.1	STRUTTURA DEL REPOSITORY GIT	5
8.2	COMPILAZIONE DEI SORGENTI E PUBBLICAZIONE DELL'IMMAGINE	5

8.3	DISPIEGAMENTO COME APPLICAZIONE STANDALONE	6
8.4	DISPIEGAMENTO SU KUBERNES	7
9	CONFIGURAZIONI DNS OPZIONALI	7
10	CARICAMENTI INIZIALI	7
11	INSTALLAZIONE FUSEKI	8
11.1	ISTRUZIONI PER INSTALLAZIONE DI FUSEKI DOCKER	8
11.2	ISTRUZIONI PER INSTALLAZIONE DI FUSEKI STANDALONE	9
11.3	ISTRUZIONI PER INSTALLAZIONE DI FUSEKI HA	10
12	GESTIONE	14

## 1 RIFERIMENTI

N.	Titolo	Autore	Versione	Data
1	PB_SGI4_MyStandard_Realizzazione 1_v1.0	EII	1.0	04/08/2020

## 2 GLOSSARIO

Termine	Descrizione

## 3 ACRONIMI

Termine	Descrizione

## 4 CONTESTO

Il documento si inquadra all'interno del progetto MyStandard un'applicazione per la gestione della Knowledge Base dei concetti del mondo MyPortal.

## 5 OBIETTIVI

Scopo del documento è fornire tutte le indicazioni necessarie al dispiegamento su Kubernetes (1.7.2 +) dei container Docker (1.12.x +) relativamente e limitatamente alle componenti di MyStandard

## 6 PREREQUISITI INFRASTRUTTURALI

La soluzione MyStandard necessita per il suo dispiegamento e la sua installazione delle seguenti componenti infrastrutturali:

- **Cluster K8s** : Kubernetes (1.7.2 +) con support di container Docker (1.12.x +)
- **Repository Nexus** : Per la pubblicazione di artefatti e immagini docker una volta compilati i sorgenti
- **Redis** : Per la gestione delle sessioni
- **MongoDB** : Per la gestione e il salvataggio delle query semantiche preconfigurate e/o predisposte dagli operatori
- **CEPH (Object Store)**: MyStandard utilizza bucket dedicati per cui può utilizzare anche un'istanza di MyBox separata dagli altri servizi MyPlace.
- **Fuseki (Persistent Store RDF)**: Richiesta la sua installazione in questo progetto
- **LogStash**: Per i log applicativi
- **ELK** : Per le ricerche full text

L'installazione e la configurazione delle componenti infrastrutturali è un prerequisito ed esula dallo scopo di documento.

## 7 PREREQUISITI SERVIZI MYPLACE E VERTICALI MYP3

La soluzione MyStandard necessita ha dipendenza verso alcuni servizi della piattaforma MyPlace:

- **MyId**: Per al gestione dell'autenticazione
- **MyProfile**: Per la gestione delle autorizzazioni e dei profili
- **MySearch** : Per le ricerche full text

MyStandard eroga ( opzionalmente ) alcune delle funzionalità attraverso API per i verticali:

- **MyIntranet** : Per l'accesso all'applicazione MyStandard da parte degli operatori degli Enti direttamente attraverso la MyIntranet.

- **MyPortal** : Per la pubblicazione e la consultazione pubblica dei cataloghi direttamente da MyPortal

MyStandard deve quindi essere raggiungibile a livello di rete da queste due applicazioni.

## 8 INSTALLAZIONE

### 8.1 STRUTTURA DEL REPOSITORY GIT

Il repository git di MyStandard ha le seguente struttura:

**/mystandard.sources**: E' la cartella che contiene i sorgenti e gli script gradle per la compilazione la creazione dell'immagine e la pubblicazione sul Repository Nexus.

**/mystandard.deploy**: E' la cartella che contiene i descrittori di base per il dispiegamento su kubernetes e gli overlay specifici per ogni ambiente target di deploy.

```
/base/  
  /external-services: si trovano i descrittori dei servizi esterni base  
  
  /ingress: si trovano i descrittori con le regole ingress base  
  
  /mystandard: si trovano i descrittori kubernetes di mystandard base  
  
/overlays/  
  
  /<overlay-env1>: Overlay specifico per environment 1  
  
  /<overlay-env2>: Overlay specifico per environment 2  
  
  /jena-docker-master : Cartella con i Docker file e le istruzioni per costruire le immagini  
  Docker di Jena Fuseki Server
```

### 8.2 COMPILAZIONE DEI SORGENTI E PUBBLICAZIONE DELL'IMMAGINE

Per la compilazione dei sorgenti, posizionarsi all'interno della directory mystandard.sources e verificare che all'interno siano presenti:

- directory mystandard-be
- directory mystandard-fe
- directory gradle/wrapper

Nel caso di ambiente Linux, eseguire quindi il comando di

```
./gradlew build
```

Nel caso di ambiente Windows, eseguire il comando di

```
gradlew.bat build
```

All'interno della directory mystandard-be/build/libs sarà presente l'artefatto compilato.

## 8.3 DISPIEGAMENTO COME APPLICAZIONE STANDALONE

Per poter eseguire lo start dell'applicazione, è importante inserire i puntamenti dei propri ambienti per i componenti da cui dipende MyStandard.

Nel caso si voglia utilizzare un nexus per scaricare o pubblicare pacchetti, modificare il file `settings.gradle` all'interno della directory `mystandard.sources` e modificare:

- parametri `gradle.ext.*` con i parametri del nexus a cui si vuole puntare

Se si vuole far partire l'applicazione nei propri ambienti, è necessario aggiungere le seguenti configurazioni:

- `mystandard-be/src/main/resources/application.yml`
  - `mystandard.owl.filename.*`: indicare il path dove trovare i file
  - `mystandard.myBoxConfigurationFilePath*`: indicare il path dove trovare il file
  - `mystandard.stateMachine`: indicare il path dove trovare il file
- `mystandard-be/src/main/resources/application-default.yml`:
  - `server.ssl.*`: se si vuole abilitare HTTPS per Spring Boot
  - `auth.fake.enabled`: se il parametro è a true, si esegue un'autenticazione fake senza invocare MyID
  - `saml.proxy.*`: per la connessione a MyID, se si è in presenza di Load Balancer o Reverse Proxy
  - `saml.key-*`: keystore con la chiave per firmare asserzioni SAML
  - `saml.app-entity-id`: Nome dell'applicazione nella configurazione per l'IDP
  - `saml.app-base-url`: URL dell'applicazione da fornire nel processo di generazione dei metadati per MyID
  - `saml.idp-metadata-url`: URL da cui ricavare i metadati IDP
  - `jwt.secret`: chiave con cui firmare i JWT
  - `myprofile.baseUrl`: url di MyProfile
  - `myprofile.fake`: se il parametro è true, non si contatta MyProfile ma si prendono le configurazioni di test del file `test_fake_users.json`
  - `mysearch.hosts`: url in formato `<host>:<port>:<protocol>` da utilizzare per contattare mySearch
  - `mysearch.entitiesMappingConfigurationAbsolutePath`: indicare il path dove trovare il file
  - `logstash.*`: riferimenti per contattare logstash
  - `mystandard.enteNazionale`: nome ente standardizzatore
  - `mystandard.fuseki-server-url`: Url in cui contattare Fuseki
  - `mongodb.*`: Riferimenti per contattare MongoDB
  - `cache.sentinel.*`: Riferimenti di Redis se si usa la modalità sentinel
  - `cache.standalone.*`: Riferimenti di Redis se si usa la modalità normale
- `mystandard-be/src/main/resources/mybox.properties`
  - `s3.*`: indicare i riferimenti a s3 se si usa s3 come Object Store
  - `swift.*`: indicare i riferimenti a SWIFT se si usa SWIFT come Object Store

Per eseguire la compilazione dei sorgenti e l'esecuzione degli stessi in un unico comando, se si è in ambiente Linux, eseguire il comando di

```
./gradlew bootRun
```

Nel caso di ambiente Windows, eseguire il comando di

```
gradlew.bat bootRun
```

Se si vuole far partire l'applicativo dall'artefatto, posizionarsi nella directory dove l'artefatto è stato generato, ed eseguire il comando di

```
java -jar <nome artefatto>.jar
```

## 8.4 DISPIEGAMENTO SU KUBERNETES

La directory base contiene le configurazioni base, che saranno poi sovrascritte dai specifici overlays definiti dentro la directory overlays.

Per il dispiegamento su un particolare environment ( per esempio "myenvironment" ) è necessario:

1. creare la relativa cartella di overlay con i puntamenti alle componenti infrastrutturali e di myplace opportunamente modificate per lo specifico ambiente.
2. Utilizzare lo strumento Kustomize per applicare le configurazioni di uno specifico ambiente ai descrittori di base.

```
a. kustomize build overlays/<myenvironment>
```

3. Installare i pod sul cluster Kubernetes con le istruzioni:

```
a. kustomize build overlays/<myenvironment> | kubectl apply -f -
```

## 9 CONFIGURAZIONI DNS OPZIONALI

E' richiesto di censire su DNS i domini scelti per l'esposizione del componente mystandard riportati di seguito in modo che risolvano su IP che viene poi rigirato verso l'ambiente di Kubernetes di deploy ed effettuare le relative regole di instradamento su INGRESS.

I domini consigliati per l'esposizione sono:

- `mystandard.collaudo.<miodominio>.it` (per ambiente di collaudo)
- `mystandard.<miodominio>.it` (per ambiente di produzione)

## 10 CARICAMENTI INIZIALI

### CEPH

E' richiesta la creazione dei bucket "MYSTD" all'interno di CEPH per essere utilizzabile mediante le **SWIFT API**

E' richiesta la creazione di un bucket "fusekimystd" all'interno di CEPH, per essere utilizzabile mediante le **S3 API**.

## MONGO

E' richiesta la creazione dell'utente e del Database su Mongo da utilizzare per MyStandard. Si possono utilizzare I seguenti comandi

```
use mystandard
```

```
db.createUser({ user: 'mystandard', pwd: 'mystandard', roles: [ { role: 'readWrite', db: 'mystandard' } ] })
```

Eseguire l'operazione di mongo import prendendo il file query.json dalla cartella di rilascio init/mongoDB

```
mongoimport -d mystandard -c query -u mystandard -p mystandard --type json --file query.json
```

## 11 INSTALLAZIONE FUSEKI

Apache Jena Fuseki è un server SPARQL che utilizza un componente TDB2 per la persistenza e le query di dati RDF.

Fuseki può essere installato come immagine docker, o come server standalone in un singolo nodo o in HA. Si descrive qui di seguito la procedura di installazione di tutte le soluzioni.

### 11.1 ISTRUZIONI PER INSTALLAZIONE DI FUSEKI DOCKER

L'obiettivo di questo paragrafo è quello di descrivere l'installazione in modalità docker e non in modalità standalone.

Per costruire l'immagine Docker di fuseki usare il Docker File:

```
/mystandard.deploy/jena-docker-master/jena-fuseki
```

Se si usa Kubernetes l'immagine Docker deve essere pushata sul repository nexus come:

```
'__YOUR_NEXUS_REPO__/fuseki:infrastructure'
```

Si sconsiglia l'uso dell'immagine Docker in ambienti di Collaudo e di Produzione.

Nel repository git sono fornite ( per ogni overlay ) anche

- una directory /init/fuseki-docker/**configuration** il cui contenuto dovrà essere montato come volume esterno all'immagine nel path /fuseki al fine di ottenere una directory /fuseki/configuration. Il volume e la directory devono avere i permessi di lettura e scrittura.
- una directory /init/fuseki-docker/**databases**, il cui contenuto dovrà essere montato come volume esterno all'immagine nel path /fuseki al fine di ottenere una directory



/fuseki/databases. Il volume e la directory devono avere i permessi di lettura e scrittura.

E' importante che solo all'init venga caricato il contenuto in quanto è oggetto di modifica lungo tutto il ciclo di vita dell'applicazione. Quindi è indispensabile che il volume non punti ad un repository GIT per non perdere le modifiche ai dati.

Verificare il corretto deploy attraverso la seguente **curl**, che venga visualizzata una tabella, che potrebbe anche non avere contenuto(<host> e <port> sono quelli dove risponde fuseki)

```
curl --data 'query=SELECT ?s ?p ?o WHERE {?s ?p ?o} LIMIT 10' --data 'output=text'
http://<host>:<port>/mystd
```

Fuseki deve essere raggiungibile da MyStandard, ed è necessario configurare l'ambiente in modo che MyStandard riesca a raggiungere il server fuseki utilizzando:

- `http://fuseki-private.collaudo.<miodominio>.it:3030` ( Collaudo )
- `http://fuseki-private.<miodominio>.it:3030` ( Produzione )

In caso le url siano diverse, è necessario rivedere la configurazione di MyStandard.

E' necessario prevedere dei meccanismi di backup per la directory **databases** montata come volume esterno.

Se si usa Kubernetes l'immagine Docker deve essere pushata sul repository nexus come:

```
'__YOUR_NEXUS_REPO__/_fuseki:infrastructure'
```

## 11.2 ISTRUZIONI PER INSTALLAZIONE DI FUSEKI STANDALONE

L'obiettivo di questo paragrafo è quello di descrivere l'installazione in modalità standalone e non in HA.

Il pacchetto che contiene il server standalone per fuseki versione 3.17 è presente all'indirizzo <http://archive.apache.org/dist/jena/binaries/apache-jena-fuseki-3.17.0.zip>

Per poter installare Fuseki come server standalone, prevedere

- JDK1.8 o successive (preferibilmente openJDK11, versione usata da MyStandard)

Per l'installazione eseguire i seguenti step:

1. Esplodere il contenuto del pacchetto in una directory che chiameremo <FUSEKI\_HOME>
2. Nel rilascio verrà fornita una directory /init/fuseki-standalone/**databases**, che dovrà essere copiata all'interno del file system in una directory che per semplicità chiameremo <FUSEKI\_DATABASE>, in questo modo si dovrà ottenere una directory con la seguente struttura <FUSEKI\_DATABASE>/**databases**. Non è vincolante che la directory <FUSEKI\_DATABASE> sia all'interno di <FUSEKI\_HOME>. Devono essere assegnati i permessi di scrittura e lettura sulla <FUSEKI\_DATABASE> per l'utente di esecuzione del comando al punto 5.
3. Accedere alla directory <FUSEKI\_HOME>, dove è presente lo script "fuseki-server"

#### 4. Eseguire il comando:

```
<FUSEKI_HOME>/fuseki-server --tdb2 --loc=<FUSEKI_MYSTD> --update /mystd
```

dove fuseki-server è lo script presente in <FUSEKI\_HOME>, mentre <FUSEKI\_MYSTD> è equivalente al path <FUSEKI\_DATABASE>/**databases/mystd**.

#### 5. Verificare il corretto deploy attraverso la seguente **curl**, che venga visualizzata una tabella, che potrebbe anche non avere contenuto(<host> e <port> sono quelli dove risponde fuseki)

```
curl --data 'query=SELECT ?s ?p ?o WHERE {?s ?p ?o} LIMIT 10' --data 'output=text'
http://<host>:<port>/mystd
```

Fuseki deve essere raggiungibile da MyStandard, ed è necessario configurare l'ambiente in modo che MyStandard riesca a raggiungere il server fuseki utilizzando:

- `http://fuseki-private.collaudo.<miodominio>.it:3030` ( Collaudo )
- `http://fuseki-private.<miodominio>.it:3030` ( Produzione )

In caso le url siano diverse, è necessario rivedere la configurazione di MyStandard.

E' necessario prevedere dei meccanismi di backup per la directory  
<FUSEKI\_DATABASE>/**databases/mystd** per garantire la persistenza dei dati.

## 11.3 ISTRUZIONI PER INSTALLAZIONE DI FUSEKI HA

L'obiettivo è descrivere l'installazione di una architettura Fuseki che supporti l'HA, e che fa affidamento sul delta patch server.

I 3 delta patch server devono utilizzare un volume Ceph per il patch Storage sul bucket fusekimystd creato in precedenza nel paragrafo dei caricamenti iniziali per CEPH.

Il pacchetto che contiene Fuseki server che supporta la HA (versione compatibile con versione 3.17 di Fuseki utilizzata da MyStandard) e che contiene il delta server, si trova all'indirizzo <https://repo1.maven.org/maven2/org/seaborne/rdf-delta/rdf-delta-dist/0.9.0/rdf-delta-dist-0.9.0.zip>

Esplodere il contenuto del pacchetto in una directory che chiameremo per semplicità <FUSEKI\_HOME>

Per poter utilizzare delta-fuseki e delta-server, prevedere

- JDK1.8 o successive (preferibilmente openJDK11, versione usata da MyStandard)

## DELTA PATCH SERVER

RDF Delta fornisce un sistema di patch log server replicato e in HA utilizzando Apache Zookeeper. Nel pacchetto `rdf-delta` scaricato precedentemente, all'interno è già fornita una istanza di Zookeeper, ma è necessario predisporre 3 istanze diverse in quanto si basa su un sistema di quorum che necessita una maggioranza di 2; è quindi richiesto di predisporre 3 installazioni.

Su ognuna delle 3 installazioni, creare una directory locale per permettere lo storage di Zookeeper, che per semplicità chiameremo `<ZKDATA>` su cui Zookeeper va a creare alcuni file interni. Nella directory devono essere assegnati i permessi di lettura e scrittura all'utente che andrà ad eseguire i comandi descritti in seguito.

Le 3 installazioni devono essere raggiunte mediante un `<host>` differente per ognuna, e devono essere raggiungibili tra di loro. Per l'ambiente di collaudo si richiede di utilizzare come host nelle 3 installazioni:

- `deltapatch1-private.collaudo.<miodominio>.it`
- `deltapatch2-private.collaudo.<miodominio>.it`
- `deltapatch3-private.collaudo.<miodominio>.it`

Per l'ambiente di produzione invece:

- `deltapatch1.<miodominio>.it`
- `deltapatch2.<miodominio>.it`
- `deltapatch3.<miodominio>.it`

In caso di utilizzo di host diversi, è necessario rivedere la configurazione dei fuseki server (descritti nel paragrafo Delta Fuseki Server).

Per l'esecuzione del delta patch server, Il comando da eseguire in ogni installazione è

```
java -jar <FUSEKI_HOME>/delta-server.jar --port 1071 --zkPort 2181 --zkData <ZKDATA> --zk=<ZK> --s3bucket=fusekimystd --s3keys=<KEYS> --s3region=<REGION> --s3endpoint=<ENDPOINT>
```

Dove i parametri sono

- **ZKDATA:** path della directory locale creata precedentemente per la memorizzazione di dati del zookeeper
- **ZK:** Zookeeper connection string contenente le info delle tre installazioni del delta server. Per l'ambiente di collaudo sarà:

```
deltapatch1-private.collaudo.<miodominio>.it:2181,  
deltapatch2-private.collaudo.<miodominio>.it:2181,  
deltapatch3-private.collaudo.<miodominio>.it:2181
```

Per l'ambiente di produzione sarà

```
deltapatch1.<miodominio>.it:2181,  
deltapatch2.<miodominio>.it:2181,  
deltapatch3.<miodominio>.it:2181
```

- **KEYS:** File property delle credenziali AWS S3, se non fornito viene utilizzato il meccanismo di autenticazione S3 predefinito standard

- **REGION:** Region su cui è configurato S3.
- **ENDPOINT:** Endpoint su cui raggiungere le API S3 di CEPH:

`<protocollo>://<hostceph_s3>:<portceph_s3>`

Verificare sulla console di vedere il LOG INFO "Delta INFO No data sources" per la conferma di una corretta installazione.

## DELTA FUSEKI SERVER

RDF Delta fornisce un server Fuseki con la variante delta che permette di comunicare ai patch server eventuali changes.

E' richiesto di predisporre 2 installazioni bilanciate dall'HA Proxy. Ognuna delle 2 installazioni deve raggiungere le 3 installazioni del Delta Patch Server descritte nel paragrafo precedente.

Per eseguire l'installazione del delta fuseki server.

- Nel rilascio verrà fornita una directory `/init/fuseki-ha/configuration` che dovrà essere copiata all'interno del file system in una directory che per semplicità chiameremo `<FUSEKI_CONF>`. in questo modo si dovrà ottenere una directory con la seguente struttura `<FUSEKI_CONF>/configuration`.
- Nel rilascio verrà fornita una directory `/init/fuseki-ha/databases` che dovrà essere copiata all'interno della directory `<FUSEKI_CONF>/configuration`, in questo modo si dovrà ottenere una directory con la seguente struttura `<FUSEKI_CONF>/configuration/databases`.
- Prendere il file `<FUSEKI_CONF>/configuration/mystd.ttl` e modificare la proprietà ed inserire la lista dei 3 delta patch server creati al punto precedente.

Modificare la proprietà **delta:changes** in questo modo:

```
o delta:changes("http://<host1>:1071/" "http://<host2>:1071/"  
"http://<host3>:1071/");
```

Eseguire il comando

```
java -jar <FUSEKI_HOME>/delta-fuseki.jar --config  
<FUSEKI_CONF>/configuration/mystd.ttl
```

Verificare il corretto deploy attraverso la seguente curl, che venga visualizzata una tabella, che potrebbe anche non avere contenuto (<host> e <port> sono quelli dove risponde il bilanciatore).

```
curl --data 'query=SELECT ?s ?p ?o WHERE {?s ?p ?o} LIMIT 10' --data  
'output=text' http://<host>:<port>/mystd
```

Fuseki deve essere raggiungibile da MyStandard, ed è necessario configurare l'ambiente in modo che MyStandard riesca a raggiungere il server fuseki tramite il bilanciatore utilizzando:

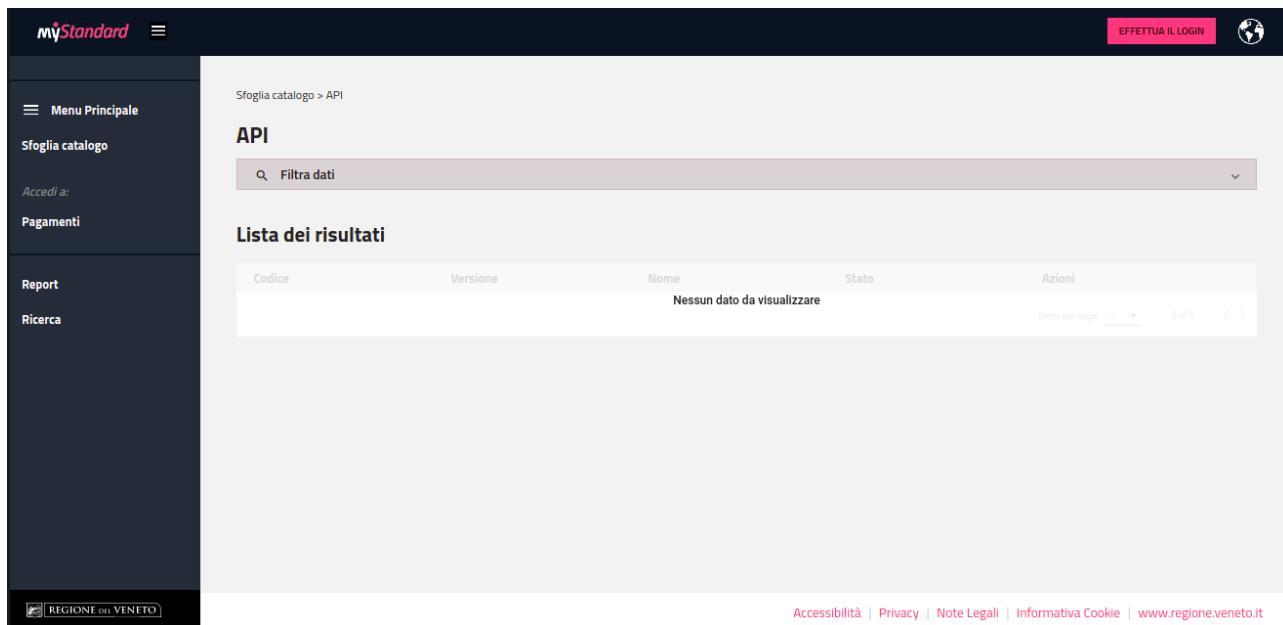
- `http://fuseki-private.collaudo.<miominio>.it:3030 ( Collaudo )`

- `http://fuseki-private.<miodominio>.it:3030` ( Produzione )

In caso le url siano diverse, è necessario rivedere la configurazione di MyStandard.

## 12 GESTIONE

Per verificare l'operatività dell'ambiente si ci deve collegare al dominio indicato precedentemente in cui verrà mostrata la home page dell'applicativo.



The screenshot shows the 'myStandard' application interface. On the left is a dark sidebar with a 'Menu Principale' and links to 'Sfoglia catalogo', 'Accedi a:', 'Pagamenti', 'Report', and 'Ricerca'. The main content area has a breadcrumb 'Sfoglia catalogo > API' and a title 'API'. Below the title is a search bar labeled 'Filtra dati'. A section titled 'Lista dei risultati' contains a table with columns: 'Codice', 'Versione', 'Nome', 'Stato', and 'Azioni'. The table body is empty, displaying the message 'Nessun dato da visualizzare'. At the bottom right of the table, it says 'Items per page: 10' and '0 of 0'. The footer of the page includes the 'REGIONE DEL VENETO' logo on the left and a row of links: 'Accessibilità', 'Privacy', 'Note Legali', 'Informativa Cookie', and 'www.regione.veneto.it' on the right.

myStandard

EFFETTUA IL LOGIN

Sfoglia catalogo > API

API

Filtra dati

Lista dei risultati

Codice	Versione	Nome	Stato	Azioni
Nessun dato da visualizzare				

Items per page: 10 0 of 0

REGIONE DEL VENETO

Accessibilità | Privacy | Note Legali | Informativa Cookie | [www.regione.veneto.it](http://www.regione.veneto.it)