

Manuale di Installazione MyPivot4

SOMMARIO

1	OBIETTIVI	2
2	PREREQUISITI INFRASTRUTTURALI.....	2
3	PREREQUISITI SERVIZI MYPLACE E VERTICALI MYP3	3
4	STRUTTURA DEL REPOSITORY	3
4.1	I SORGENTI.....	4
5	INSTALLAZIONE E CONFIGURAZIONE.....	5
5.1	PARAMETRI PER IL RECUPERO DEL SORGENTE	5
5.2	PARAMETRI DI INSTALLAZIONE	5
5.2.1	<i>Valori ambiente di test</i>	<i>15</i>
5.3	ISTRUZIONI DI INSTALLAZIONE STANDALONE.....	26
5.3.1	<i>Installazione back-end (Api Java Spring Boot).....</i>	<i>26</i>
5.3.2	<i>Installazione front-end (Angular).....</i>	<i>27</i>
5.4	INSTALLAZIONE PER DEPLOYMENT IN AMBIENTE KUBERNATES	29
6	CARICAMENTI INIZIALI.....	29
6.1	IMPIANTO DA SISTEMA PREESISTENTE IN UNA VERSIONE PRECEDENTE	29
6.2	IMPIANTO EXNOVO SENZA SISTEMA PREESISTENTE	30
7	INSTALLAZIONE JOB.....	30

1 OBIETTIVI

Scopo del documento è fornire tutte le indicazioni necessarie al dispiegamento standalone o in alternativa su Kubernetes con container Docker relativamente e limitatamente alle componenti di MyPivot4.

2 PREREQUISITI INFRASTRUTTURALI

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

- **Redis** : Per la gestione delle sessioni
- **PostgreSQL**: per la memorizzazione dei dati applicativi
- **Server di Posta**: per l'invio di email ai vari utenti del sistema

In caso di installazione su K8s (si veda il documento MI) sono necessari anche:

- **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
- **CEPH** (Object Store) per lo storage di file applicativi

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

3 PREREQUISITI SERVIZI MYPLACE E VERTICALI MYP3

La soluzione MyPivot4 ha dipendenza verso alcuni servizi della piattaforma MyPlace di MyPortal3:

- **MyProfile**: per la gestione delle autorizzazioni e dei profili

4 STRUTTURA DEL REPOSITORY

Il repository git di MyPivot ha la seguente struttura:

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

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

MyPivot4 è suddiviso nelle seguenti componenti:

- applicazione di front-end "MyPivot4-fe" basato su tecnologia Angular;
- applicazione di back-end "MyPivot4-be" realizzato in tecnologia Java;
- diversi batch scritti in linguaggio Java.

4.1 I SORGENTI

La cartella mypivot.sources contiene i sorgenti dell'applicazione ed è così strutturata:

- gradle: contiene file utilizzati per la build tramite
- Gradle myPivot-batch: contiene i sorgenti dei processi batch Talend
- mypivot-db: contiene i sorgenti degli script database
- mypivot4-be: contiene i sorgenti del back-end Java Spring Boot myPivot Operatore
- mypivot4-fe: contiene i sorgenti del front-end Angular myPivot Operatore

5 INSTALLAZIONE E CONFIGURAZIONE

5.1 PARAMETRI PER IL RECUPERO DEL SORGENTE

La compilazione dell'applicazione di front-end avviene mediante npm, mentre per il back-end viene utilizzato lo strumento gradle. I batch vengono invece consegnati con artefatti già compilati in quanto sono principalmente realizzati mediante lo strumento Talend.

Scopo del documento è fornire tutte le indicazioni necessarie al dispiegamento su infrastruttura standard non Dockerizzata su macchine fisiche/Virtuali limitatamente alle componenti di MyPivot4

- MyPivot4-fe
- MyPivot4-be
- MyPivot4-batch
- MyPivot4-db

È propedeutica la disponibilità dei seguenti componenti della piattaforma MyPlace:

- MyProfile

e richiede i seguenti componenti infrastrutturali:

- Server AMQP: **Artemis v2.17.0**
- Piattaforma per Monitoraggio applicativo: **OpenDistro v1.11.0**
- Monitoraggio cluster: **Rancher v2.5.0**
- Agent per il logging: **Fluentd v1.4.2-debian-elasticsearch-1.1**
- Ingress controller: **Nginx nginx-ingress-controller:0.32.0**
- Caching e gestore delle sessioni: **redis: 6.0.9-debian-10-r13**
- Volume condiviso (CEPHFS)

Database Relazionale:

- PostgreSQL 11 o superiore

5.2 PARAMETRI DI INSTALLAZIONE

Si elencano di seguito i parametri applicativi soggetti a variazione dipendente dall'installazione (test, collaudo, produzione). Indicare l'utilizzo del parametro nei vari file, indicando il parametro tra parentesi quadre.

Codice parametro	Descrizione	File interessati e indicazione puntale (linea o parametro)
spring.datasource.pivot.url	stringa di connessione al DB "PIVOT" Maggiori info al link: https://docs.spring.io/spring-boot/docs/2.4.2/reference/html/appendix-application-properties.html#common-application-properties-data	file: application.properties parametro: spring.datasource.pivot.url
spring.datasource.pivot.driverClassName	driver JDBC "PIVOT" Maggiori info al link: https://docs.spring.io/spring-boot/docs/2.4.2/reference/html/appendix-application-properties.html#common-application-properties-data	file: application.properties parametro: spring.datasource.pivot.driverClassName
spring.datasource.pivot.username	username connessione al DB "PIVOT" Maggiori info al link: https://docs.spring.io/spring-boot/docs/2.4.2/reference/html/appendix-application-properties.html#common-application-properties-data	file: application.properties parametro: spring.datasource.pivot.username
spring.datasource.pivot.password	password connessione al DB "PIVOT"	file: application.properties parametro: spring.datasource.pivot.password
spring.datasource.pivot.minimumIdle	num. min. connessioni idle ad DB "PIVOT"	file: application.properties parametro: spring.datasource.pivot.minimumIdle
spring.datasource.pivot.maximumPoolSize	dimensione max. del pool di connessione al DB "PIVOT"	file: application.properties parametro: spring.datasource.pivot.maximumPoolSize
jdbc.limit.default	numero massimo di risultati ritornati dalle query di ricerca di MyPivot	file: application.properties parametro: jdbc.limit.default
server.error.include-message	nel caso di errori di sistema durante l'esecuzione di servizi Rest, mostra o meno l'errore nella response JSON (valore "always" -> mostra, "never" -> nascondi)	file: application.properties parametro: server.error.include-message
spring.application.name	nome applicazione	file: application.properties parametro: spring.application.name
logging.level.root	Livello base di logging	file: application.properties parametro: logging.level.root

logging.level.it.regioneveneto.mygov.payment	<p>Livello di logging per il package java "it.regioneveneto.mygov.payment"</p> <p>(è possibile impostare, secondo le necessità e per ottimizzare la scrittura dei log, un livello di log custom per ogni package/classe java impostando un nuovo parametro con chiave "logging.level.[package / classe]" e valore il livello di logging desiderato)</p>	<p>file: application.properties</p> <p>parametro: logging.level.it.regioneveneto.mygov.payment</p>
mypay4.logging.stacktrace.file	<p>pattern degli stack-trace degli errori, loggati nei file di log</p> <p>(il valore suggerito è ottimizzato per l'importazione dei file in ELK)</p>	<p>file: application.properties</p> <p>parametro: mypay4.logging.stacktrace.file</p>
mypay4.logging.stacktrace.console	<p>pattern degli stack-trace degli errori, loggati nella console</p> <p>(il valore suggerito è facilmente leggibile, ma poco gestibile da tool di gestione log quali ELK)</p>	<p>file: application.properties</p> <p>parametro: mypay4.logging.stacktrace.console</p>
logging.pattern.console	pattern dei messaggi di log nei file di log	<p>file: application.properties</p> <p>parametro: logging.pattern.console</p>
logging.pattern.file	pattern dei messaggi di log nella console	<p>file: application.properties</p> <p>parametro: logging.pattern.file</p>
logging.level.org.springframework.web.filter.CommonsRequestLoggingFilter	Livello di logging per il package java "org.springframework.web.filter.CommonsRequestLoggingFilter"	<p>file: application.properties</p> <p>parametro: logging.level.org.springframework.web.filter.CommonsRequestLoggingFilter</p>
method-execution-logging.enabled	<p>Abilita il log dell'esecuzione dei metodi; valori possibili:</p> <ul style="list-style-type: none"> - false: disabilitato completamente - method: logging breve (no parametri in/out) - true: log completo 	<p>file: application.properties</p> <p>parametro: method-execution-logging.enabled</p>

request-logging.enabled	abilita il log di ogni request http (incluso headers + payload); valori possibili: true / false	file: application.properties parametro: request-logging.enabled
datasource-pool-logging.enabled	abilita il log del datasource connection pool; valori possibili: true / false	file: application.properties parametro: datasource-pool-logging.enabled
sql-logging.enabled	abilita il log degli statement SQL; valori possibili: - false: disabilitato - short: logging breve (no parametri in/out) - true: log completo	file: application.properties parametro: sql-logging.enabled
jwt.secret	secret usato per generare i token JWT (per generare un nuovo secret random, si può usare la classe it.regioneveneto.mygov.payment.MyPivot4.security.JwtTokenUtil)	file: application.properties parametro: jwt.secret
jwt.validity.seconds	validita dei token JWT in secondi	file: application.properties parametro: jwt.validity.seconds
jwt.usage-check.grace-period.milliseconds	se verifica "usage-check" abilitata, vengono accettati i token già usati da meno di questo valore (in milliseconds)	file: application.properties parametro: jwt.usage-check.grace-period.milliseconds
jwt.usage-check.enabled	se abilitato (valore true), i token possono essere usati una sola volta e poi vengono invalidati (c'è un periodo di tolleranza per cui i token usati sono ancora validi i milliseconds espressi nel parametro "jwt.usage-check.grace-period.milliseconds"); valori possibili: true/false (nota: se si imposta questo valore a true, é necessario porre a true anche il parametro jwt.rolling-token.enabled)	file: application.properties parametro: jwt.usage-check.enabled
jwt.usage-check.ignorelongcall.milliseconds	se non vuoto, ignora chiamate piu lunghe del valore soglia (in milliseconds) specificato da questo parametro, nel controllo di "usage-check" dei token	file: application.properties parametro: jwt.usage-check.ignorelongcall.milliseconds
jwt.rolling-token.enabled	se abilitato (valore true), ad ogni response dei servizi rest viene fornito un nuovo token JWT (in un header); valori possibili: true/false	file: application.properties parametro: jwt.rolling-token.enabled

cache.type	tipologia del server Redis utilizzato; valori possibili: - sentinel - cluster - standalone	file: application.properties parametro: cache.type
cache.sentinelMaster	usato solo se cache.type=sentinel; server master	file: application.properties parametro: cache.sentinelMaster
cache.sentinelNodes	usato solo se cache.type=sentinel; lista, separati da ",", dei nodi nel formato "hostname:porta"	file: application.properties parametro: cache.sentinelNodes
cache.clusterNodes	usato solo se cache.type=cluster; lista, separati da ",", dei nodi nel formato "hostname:porta"	file: application.properties parametro: cache.clusterNodes
cache.clusterMaxRedirects	usato solo se cache.type=cluster; numero max redirect	file: application.properties parametro: cache.clusterMaxRedirects
cache.useEmbeddedRedis	usato solo se cache.type=standalone e in ambienti di sviluppo; se true, viene lanciato e usato un server redis locale al lancio dell'applicazione Spring Boot	file: application.properties parametro: cache.useEmbeddedRedis
cache.standaloneHost	usato solo se cache.type=standalone; host del server standalone Redis	file: application.properties parametro: cache.standaloneHost
cache.standalonePort	usato solo se cache.type=standalone; porta del server standalone Redis	file: application.properties parametro: cache.standalonePort
cache.password	usato solo se cache.type=standalone; password del server standalone Redis	file: application.properties parametro: cache.password
cache.cachePrefix	a tutte le cache definite nell'applicazione viene aggiunto come prefisso il valore di questo parametro; utile nel caso di server Redis condiviso da piu applicazioni, per evitare collisioni nei nomi delle cache	file: application.properties parametro: cache.cachePrefix
cache.timeoutSeconds	timeout di default delle cache in secondi; dopo questo numero di secondi le cache vengono invalidate	file: application.properties parametro: cache.timeoutSeconds
cache.cacheExpirations.jwtTokenUsageCache	esempio di come definire un timeout custom (rispetto a quello di default) per una cache specifica (in questo caso la cache "jwtTokenUsageCache"; é possibile definire un timeout custom per qualsiasi cache impostando il parametro "cache.cacheExpiration.[NOME CACHE])	file: application.properties parametro: cache.cacheExpirations.jwtTokenUsageCache

cache.cacheExpirations.uploadCache	esempio di come definire un timeout custom (rispetto a quello di default) per una cache specifica (in questo caso la cache "uploadCache")	file: application.properties parametro: cache.cacheExpirations.uploadCache
upload.max-size-MiB	massima dimensione per l'upload di un file (in MiB)	file: application.properties parametro: upload.max-size-MiB
spring.servlet.multipart.max-file-size	massima dimensione per l'upload di un file (Spring servlet multipart)	file: application.properties parametro: spring.servlet.multipart.max-file-size
spring.servlet.multipart.max-request-size	massima dimensione per una request multipart (Spring servlet multipart)	file: application.properties parametro: spring.servlet.multipart.max-request-size
export.max-records	Massimo numero di righe affinché la funzionalità di export sia abilitata (su visualizza riconciliazioni / anomalie)	file: application.properties parametro: export.max-records
server.forward-headers-strategy	strategia di gestione degli headers X-Forwarded-*	file: application.properties parametro: server.forward-headers-strategy
server.use-forward-headers	abilita la gestione degli headers X-Forwarded-*	file: application.properties parametro: server.use-forward-headers
server.ssl.enabled	abilita protocollo SSL in Spring Boot	file: application.properties parametro: server.ssl.enabled
server.port	porta su cui Spring Boot espone la webapp	file: application.properties parametro: server.port
server.http.port	se SSL abilitato e si vuole esporre anche HTTP, indicare la porta HTTP; 0 per non abilitare HTTP in questo caso	file: application.properties parametro: server.http.port
server.ssl.key-store	se SSL abilitato in Spring Boot, path del key-store	file: application.properties parametro: server.ssl.key-store
server.ssl.key-store-password	se SSL abilitato in Spring Boot, password del key-store	file: application.properties parametro: server.ssl.key-store-password
server.ssl.key-alias	se SSL abilitato in Spring Boot, alias della chiave	file: application.properties parametro: server.ssl.key-alias
server.ssl.key-password	se SSL abilitato in Spring Boot, password della chiave	file: application.properties parametro: server.ssl.key-password
server.servlet.context-path	context path dell'applicazione	file: application.properties parametro: server.servlet.context-path
server.error.path	path del controller di errore di default	file: application.properties parametro: server.error.path
cors.enabled	se true, è abilitato il CORS (necessario se front-end angular e API REST sono esposte su host/porte differenti)	file: application.properties parametro: cors.enabled

static.serve.enabled	se true, è abilitata l'esposizione di contenuti statici su Spring Boot (utilizzato solo in ambiente di sviluppo)	file: application.properties parametro: static.serve.enabled
static.serve.paths	se abilitata l'esposizione di contenuti statici, path su cui sono esposti (se non è abilitata l'esposizione di contenuti statici (static.serve.enabled=false) questo valore può rimanere vuoto)	file: application.properties parametro: static.serve.paths
static.serve.locations	se abilitata l'esposizione di contenuti statici, cartelle su cui sono disponibili i contenuti da esporre (se non è abilitata l'esposizione di contenuti statici (static.serve.enabled=false) questo valore può rimanere vuoto)	file: application.properties parametro: static.serve.locations
app.be.absolute-path	url pubblica (nel senso: accessibile direttamente dal browser degli utenti) per accedere alle API di MyPivot	file: application.properties parametro: app.be.absolute-path
app.fe.absolute-path	url pubblica (nel senso: accessibile direttamente dal browser degli utenti) per accedere alla webapp di MyPivot	file: application.properties parametro: app.fe.absolute-path
ws.wSDL	indica se i WSDL esposti sono statici o generati dinamicamente; valori possibili: static/dynamic	file: application.properties parametro: ws.wSDL
auth.fake.enabled	se true, è abilitato un finto meccanismo di login che accetta qualsiasi password (da utilizzare esclusivamente in ambiente di sviluppo)	file: application.properties parametro: auth.fake.enabled
pivot.identificativoIntermediarioPA	codice ipa ente intermediario dei pagamenti di default	file: application.properties parametro: pivot.identificativoIntermediarioPA
pivot.identificativoIntermediarioPA.name	denominazione ente intermediario dei pagamenti di default	file: application.properties parametro: pivot.identificativoIntermediarioPA.name
mybox.path.root	path assoluto su file system della cartella root utilizzata per lo storage dei file su volume condiviso (CEPHFS)	file: application.properties parametro: mybox.path.root
mypay.path.relative.data	path relativo della cartella root di MyPivot nell'ambito dello storage dei file	file: application.properties parametro: mypay.path.relative.data
mypay.path.manage	path assoluto della cartella "MANAGE"	file: application.properties parametro: mypay.path.manage

mypay.path.manage.log	path assoluto della cartella "MANAGE/log"	file: application.properties parametro: mypay.path.manage.log
block.whitelist.payercf	Elenco (separato da virgola) di codici fiscali che saranno gli unici su cui non saranno bloccate operazioni di pagamento PagoPA. Se vuoto, nessun blocco è attivato.	file: application.properties parametro: block.whitelist.payercf
block.blacklist.payercf	Elenco (separato da virgola) di codici fiscali su cui saranno bloccate le operazioni di pagamento PagoPA. Se vuoto, nessun blocco è attivato.	file: application.properties parametro: block.blacklist.payercf
block.whitelist.operation	Elenco (separato da virgola) di primitive PagoPA che saranno le uniche che non verranno bloccate. Se vuoto, nessun blocco è attivato.	file: application.properties parametro: block.whitelist.operation
block.blacklist.operation	Elenco (separato da virgola) di primitive PagoPA che saranno bloccate. Se vuoto, nessun blocco è attivato.	file: application.properties parametro: block.blacklist.operation
block.errorMessage	Messaggio di errore ritornato al chiamante quando i blocchi di operazioni/codici fiscali bloccano l'esecuzione di una request.	file: application.properties parametro: block.errorMessage
spring.artemis.host	Host del broker Artemis	file: application.properties parametro: spring.artemis.host
spring.artemis.port	Porta del broker Artemis	file: application.properties parametro: spring.artemis.port
spring.artemis.user	Username di login del broker Artemis	file: application.properties parametro: spring.artemis.user
spring.artemis.password	Password di login del broker Artemis	file: application.properties parametro: spring.artemis.password
spring.jms.cache.producers	Se true, è abilitata la caching dei producers	file: application.properties parametro: spring.jms.cache.producers
queue.export-pagati	nome della coda utilizzata nella funzionalità "export pagati"	file: application.properties parametro: queue.export-pagati
queue.rendicontazione-standard	nome della coda utilizzata nella funzionalità "rendicontazione standard"	file: application.properties parametro: queue.rendicontazione-standard
queue.tesoreria	nome della coda utilizzata nella funzionalità "tesoreria"	file: application.properties parametro: queue.tesoreria
queue.export-dovuti	nome della coda utilizzata nella funzionalità "export dovuti"	file: application.properties parametro: queue.export-dovuti
flussiexport.import-path	Nome della cartella utilizzata per l'upload dei file nella funzionalità "flussi export"	file: application.properties parametro: flussiexport.import-path
rendicontazione.import-path	Nome della cartella utilizzata per l'upload dei file nella funzionalità "flussi rendicontazione"	file: application.properties parametro: rendicontazione.import-path

tesoreria.import-path	Nome della cartella utilizzata per l'upload dei file nella funzionalità "flussi tesoreria"	file: application.properties parametro: tesoreria.import-path
flussoposte.import-path	Nome della cartella utilizzata per l'upload dei file nella funzionalità "flussi poste"	file: application.properties parametro: flussoposte.import-path
springdoc.api-docs.enabled	se true, è abilitato l'endpoint con la documentazione SpringDoc autogenerata per le API REST	file: application.properties parametro: springdoc.api-docs.enabled
springdoc.api-docs.path	path (relativo al context-root della webapp SpringBoot) dell'endpoint SpringDoc	file: application.properties parametro: springdoc.api-docs.path
springdoc.swagger-ui.enabled	se true, è abilitato l'endpoint del front-end Swagger-UI	file: application.properties parametro: springdoc.swagger-ui.enabled
springdoc.swagger-ui.path	path (relativo al context-root della webapp SpringBoot) del front-end Swagger-UI	file: application.properties parametro: springdoc.swagger-ui.path
springdoc.swagger-ui.tagsSorter	configurazione dell'ordinamento dei tag delle API	file: application.properties parametro: springdoc.swagger-ui.tagsSorter
springdoc.swagger-ui.operationsSorter	configurazione dell'ordinamento delle operazioni delle API	file: application.properties parametro: springdoc.swagger-ui.operationsSorter
springdoc.swagger-ui.showCommonExtensions	controllo della visualizzazione dei parametri delle API	file: application.properties parametro: springdoc.swagger-ui.showCommonExtensions
management.endpoints.enabled-by-default	endpoint di monitoraggio applicativo: se true, sono tutti abilitati di default	file: application.properties parametro: management.endpoints.enabled-by-default
management.endpoint.health.enabled	endpoint di monitoraggio "health": se true, è abilitato	file: application.properties parametro: management.endpoint.health.enabled
management.endpoint.info.enabled	endpoint di monitoraggio "info": se true, è abilitato	file: application.properties parametro: management.endpoint.info.enabled
management.endpoints.web.exposure.include	endpoint di monitoraggio da includere (* indica tutti)	file: application.properties parametro: management.endpoints.web.exposure.include
management.endpoint.health.show-details	endpoint di monitoraggio "health": determina se sono mostrati o meno tutti i dettagli	file: application.properties parametro: management.endpoint.health.show-details
management.endpoint.health.probes.enabled	endpoint di monitoraggio "health": abilita le sonde "liveness" e "readiness"	file: application.properties parametro: management.endpoint.health.probes.enabled

spring.jackson.serialization.FAIL_ON_EMPTY_BEANS	configurazione di Jackson nel caso in cui si prova a serializzare un bean senza alcun "accessor"	file: application.properties parametro: spring.jackson.serialization.FAIL_ON_EMPTY_BEANS
saml.key-store	keystore usato per l'interazione con MyId	file: application.properties parametro: saml.key-store
saml.key-store-password	password del keystore usato per l'interazione con MyId	file: application.properties parametro: saml.key-store-password
saml.key-alias	alias della chiave usata per l'interazione con MyId	file: application.properties parametro: saml.key-alias
saml.key-password	password della chiave usata per l'interazione con MyId	file: application.properties parametro: saml.key-password
saml.app-base-url	base url di MyPivot4 per identificazione verso MyId	file: application.properties parametro: saml.app-base-url
saml.app-entity-id	entity-id definita sulla configurazione di MyId per MyPivot4	file: application.properties parametro: saml.app-entity-id
saml.idp-metadata-url	url su cui MyId espone i suoi metadati	file: application.properties parametro: saml.idp-metadata-url
saml.idp-metadata-https-cert	se MyId non fa uso di un certificato HTTPS valido, path del file contenente il certificato di MyId (necessario per il suo trust)	file: application.properties parametro: saml.idp-metadata-https-cert
saml.idp-selection-path	parametro di configurazione interno di MyId	file: application.properties parametro: saml.idp-selection-path
saml.failure-url	parametro di configurazione interno di MyId	file: application.properties parametro: saml.failure-url
saml.logout-url	parametro di configurazione interno di MyId	file: application.properties parametro: saml.logout-url
saml.entry.logout	parametro di configurazione interno di MyId	file: application.properties parametro: saml.entry.logout
saml.entry.metadata	parametro di configurazione interno di MyId	file: application.properties parametro: saml.entry.metadata
saml.entry.login	parametro di configurazione interno di MyId	file: application.properties parametro: saml.entry.login
saml.entry.SSO	parametro di configurazione interno di MyId	file: application.properties parametro: saml.entry.SSO
saml.entry.SSOHoK	parametro di configurazione interno di MyId	file: application.properties parametro: saml.entry.SSOHoK
saml.entry.SingleLogout	parametro di configurazione interno di MyId	file: application.properties parametro: saml.entry.SingleLogout
saml.entry.discovery	parametro di configurazione interno di MyId	file: application.properties parametro: saml.entry.discovery

saml.proxy.enabled	se true, abilita la configurazione nei casi in cui MyPivot4 è dietro un reverse proxy	file: application.properties parametro: saml.proxy.enabled
saml.proxy.scheme	protocollo di come è esposta MyPivot4 dietro il reverse proxy	file: application.properties parametro: saml.proxy.scheme
saml.proxy.server-name	server name di come è esposta MyPivot4 dietro il reverse proxy	file: application.properties parametro: saml.proxy.server-name
saml.proxy.server-port	porta di come è esposta MyPivot4 dietro il reverse proxy	file: application.properties parametro: saml.proxy.server-port
saml.proxy.include-port	se true, è inclusa la porta	file: application.properties parametro: saml.proxy.include-port
saml.proxy.context-path	context-path di come è esposta MyPivot4 dietro il reverse proxy	file: application.properties parametro: saml.proxy.context-path
myprofile.baseUrl	base url delle api di MyProfile	file: application.properties parametro: myprofile.baseUrl
myprofile.applCode	il codice dell'applicazione di MyPivot in MyProfile (è usato solo se il sistema di profilazione esterna è usato ed è MyProfile)	file: application.properties parametro: myprofile.applCode
mypivot.codIpaEntePredefinito	Il codice dell'ente predefinito	file: application.properties parametro: mypivot.codIpaEntePredefinito
a2a.mypay.private	Chiave RSA privata(codificata base 64) del sistema "MyPivot4" usato per l'invocazione "application 2 application" di API di MyPay. Per un esempio di come generare la chiave si può consultare il metodo "main" della classe it.regioneveneto.mygov.payment.mypay4.security.JwtTokenUtil	file: application.properties parametro: a2a.mypay.private
a2a.mypay.baseUrl	Base url del sistema "MyPay4" (API backend) usato per l'invocazione "application 2 application" di API di MyPay.	file: application.properties parametro: a2a.mypay.baseUrl

5.2.1 Valori ambiente di test

Codice parametro	Valore attuale	Valore consigliato (se [SERVIZIO])
------------------	----------------	------------------------------------

spring.datasource.pivot.url	[SERVIZIO]	
spring.datasource.pivot.driverClassName	org.postgresql.Driver	
spring.datasource.pivot.username	[SERVIZIO]	
spring.datasource.pivot.password	[SERVIZIO]	
spring.datasource.pivot.minimumIdle	[SERVIZIO]	5
spring.datasource.pivot.maximumPoolSize	[SERVIZIO]	10
jdbc.limit.default	500	
server.error.include-message	always	
spring.application.name	mypivot4	
logging.level.root	[SERVIZIO]	INFO

logging.level.it.regioneveneto.mygov.payment	[SERVIZIO]	DEBUG
mypay4.logging.stacktrace.file	%replace(%xEx){'\n','\u2028'}%nopex%n	
mypay4.logging.stacktrace.console	[SERVIZIO]	%n%xEx
logging.pattern.console	[SERVIZIO]	%clr(%d{yyyy-MM-dd HH:mm:ss.SSS}){faint} %clr(user=%X{user:--} level=%p marker=%replace(%marker){'^\$', '-'}) %clr(\${PID:- }){magenta} %clr(---){faint} %clr([%t]){faint} %clr(%logger{39}){cyan} %clr(:){faint} %m\${mypay4.logging.stacktrace.console}
logging.pattern.file	%d{yyyy-MM-dd HH:mm:ss.SSS} user=%X{user:--} level=%p marker=%replace(%marker){'^\$', '-'} \${PID:- } --- [%t] %logger{39} : %m\${mypay4.logging.stacktrace.file}	
logging.level.org.springframework.web.filter.CommonsRequestLoggingFilter	[SERVIZIO]	DEBUG
method-execution-logging.enabled	[SERVIZIO]	method

request-logging.enabled	[SERVIZIO]	false
datasource-pool-logging.enabled	[SERVIZIO]	false
sql-logging.enabled	[SERVIZIO]	short
jwt.secret	[SERVIZIO]	
jwt.validity.seconds	1800	
jwt.usage-check.grace-period.milliseconds	3000	
jwt.usage-check.enabled	true	
jwt.usage-check.ignorelongcall.milliseconds	10000	
jwt.rolling-token.enabled	true	
cache.type	[SERVIZIO]	

cache.sentinelMaster	[SERVIZIO]	
cache.sentinelNodes	[SERVIZIO]	
cache.clusterNodes	[SERVIZIO]	
cache.clusterMaxRedirects	[SERVIZIO]	3
cache.useEmbeddedRedis	false	
cache.standaloneHost	[SERVIZIO]	
cache.standalonePort	[SERVIZIO]	
cache.password	[SERVIZIO]	
cache.cachePrefix	[SERVIZIO]	MyPivot4
cache.timeoutSeconds	[SERVIZIO]	900
cache.cacheExpirations.jwtTokenUsageCache	[SERVIZIO]	\${jwt.validity.seconds}
cache.cacheExpirations.uploadCache	[SERVIZIO]	\${cache.timeoutSeconds}
upload.max-size-MiB	10	

spring.servlet.multipart.max-file-size	10MB	
spring.servlet.multipart.max-request-size	10MB	
export.max-records	[SERVIZIO]	100000
server.forward-headers-strategy	framework	
server.use-forward-headers	[SERVIZIO]	true
server.ssl.enabled	[SERVIZIO]	false
server.port	[SERVIZIO]	
server.http.port	[SERVIZIO]	
server.ssl.key-store	[SERVIZIO]	
server.ssl.key-store-password	[SERVIZIO]	
server.ssl.key-alias	[SERVIZIO]	
server.ssl.key-password	[SERVIZIO]	
server.servlet.context-path	[SERVIZIO]	
server.error.path	/failure	
cors.enabled	false	
static.serve.enabled	false	

static.serve.paths	[VUOTO]	
static.serve.locations	[VUOTO]	
app.be.absolute-path	[SERVIZIO]	
app.fe.absolute-path	[SERVIZIO]	
ws.wsdl	static	
auth.fake.enabled	false	
pivot.identificativoIntermediarioPA	[SERVIZIO]	
pivot.identificativoIntermediarioPA.name	[SERVIZIO]	
mybox.path.root	[SERVIZIO]	/opt/E45
mypay.path.relative.data	DATA	
mypay.path.manage	\${mybox.path.root}/MANAGE	
mypay.path.manage.log	\${mypay.path.manage}/log	
block.whitelist.payercf	[VUOTO]	

block.blacklist.payercf	[VUOTO]	
block.whitelist.operation	[VUOTO]	
block.blacklist.operation	[VUOTO]	
block.errorMessage	[VUOTO]	
spring.artemis.host	[SERVIZIO]	
spring.artemis.port	[SERVIZIO]	
spring.artemis.user	[SERVIZIO]	
spring.artemis.password	[SERVIZIO]	
spring.jms.cache.producers	false	
queue.export-pagati	EXPORT_PAGATI	
queue.rendicontazione-standard	RENDICONTAZIONE_STANDARD	
queue.tesoreria	TESORERIA	
queue.export-dovuti	EXPORT_DOVUTI	
flussiexport.import-path	/EXPORT_PAGATI/	
rendicontazione.import-path	/RENDICONTAZIONE_FLUSSI_MP/WEB/S TANDARD/	
tesoreria.import-path	/TESORERIA_FLUSSI_MP/	
flussoposte.import-path	/TESORERIA_FLUSSI_MP/WEB/POSTE/	
springdoc.api-docs.enabled	[SERVIZIO]	true

springdoc.api-docs.path	[SERVIZIO]	/public/api-docs
springdoc.swagger-ui.enabled	[SERVIZIO]	true
springdoc.swagger-ui.path	[SERVIZIO]	/public/swagger-ui.html
springdoc.swagger-ui.tagsSorter	[SERVIZIO]	alpha
springdoc.swagger-ui.operationsSorter	[SERVIZIO]	alpha
springdoc.swagger-ui.showCommonExtensions	[SERVIZIO]	true
management.endpoints.enabled-by-default	[SERVIZIO]	false
management.endpoint.health.enabled	[SERVIZIO]	true
management.endpoint.info.enabled	[SERVIZIO]	true
management.endpoints.web.exposure.include	[SERVIZIO]	*
management.endpoint.health.show-details	[SERVIZIO]	always
management.endpoint.health.probes.enabled	[SERVIZIO]	true
spring.jackson.serialization.FAIL_ON_EMPTY_BEANS	false	
saml.key-store	[SERVIZIO]	
saml.key-store-password	[SERVIZIO]	
saml.key-alias	[SERVIZIO]	

saml.key-password	[SERVIZIO]	
saml.app-base-url	[SERVIZIO]	\${app.be.absolute-path}
saml.app-entity-id	[SERVIZIO]	\${app.be.absolute-path}
saml.idp-metadata-url	[SERVIZIO]	[MY_ID_METADATA_URL]
saml.idp-metadata-https-cert	[SERVIZIO]	[VUOTO]
saml.idp-selection-path	/saml/idpSelection	
saml.failure-url	/failure	
saml.logout-url	/index.html	
saml.entry.logout	/saml/logout/**	
saml.entry.metadata	/saml/metadata/**	
saml.entry.login	/saml/login/**	
saml.entry.SSO	/saml/SSO/**	
saml.entry.SSOHoK	/saml/SSOHoK/**	
saml.entry.SingleLogout	/saml/SingleLogout/**	
saml.entry.discovery	/saml/discovery/**	
saml.proxy.enabled	[SERVIZIO]	
saml.proxy.scheme	[SERVIZIO]	
saml.proxy.server-name	[SERVIZIO]	
saml.proxy.server-port	[SERVIZIO]	
saml.proxy.include-port	[SERVIZIO]	

saml.proxy.context-path	[SERVIZIO]	
myprofile.baseUrl	[SERVIZIO]	(dovrebbe puntare all'istanza di MyProfile già presente in MyPortal3)
myprofile.applCode	MY_PIVOT	
mypivot.codIpaEntePredefinito	[CODICE_IPA_ENTE_PREDEFINITO]	
a2a.mypay.private	[SERVIZIO]	
a2a.mypay.baseUrl	[SERVIZIO]	(dovrebbe puntare alla context root dell'app Spring Boot che espone le API operatore di MyPay4; essendo una comunicazione server-2-server, è consigliabile per motivi di performance utilizzare l'indirizzo "interno" e non quello "pubblico" esposto su internet, compatibilmente con la configurazione di rete adottata)

Legenda Stato :

NUOVO: inserito in questo documento;

MODIFICATO: modificato rispetto alla versione precedente;

STABILE: presente identico in precedente versione o in documenti precedenti relativi a catena analoga.

SERVIZIO: indica che il valore deve essere impostato dal servizio di gestione applicativa sull'ambiente.

5.3 ISTRUZIONI DI INSTALLAZIONE STANDALONE

5.3.1 Installazione back-end (Api Java Spring Boot)

Prerequisiti:

- OpenJDK v11 (LTS) [scaricabile da <https://adoptium.net/>]
- Client Git (con accesso ad internet) sulla macchina in cui si effettua la build
- I componenti e i servizi esterni necessari per l'utilizzo di MyPivot siano attivi e correttamente configurati:
 - Database PostgreSQL
 - MyProfile (autorizzazione)
 - Redis Cache
 - ActiveMQ
- Le indicazioni e i comandi in seguito presuppongono l'interazione con una shell Linux; sebbene sia possibile utilizzare altri S.O., è considerato fuori dallo scopo di questo documento l'adattamento dei comandi / script indicati agli altri S.O.

Procedura di build:

1. Scaricare i sorgenti dal repository di riferimento (la cartella in cui sono scaricati i sorgenti sarà indicata nel seguito come \$PRJ_ROOT)
2. Effettuare il checkout del tag GIT su cui effettuare la build / deploy
3. Posizionarsi nella root del progetto "backend" (path: \$PRJ_ROOT/mypivot4-be)
4. Lanciare la build Gradle con il comando:

```
./gradlew clean build -PSERVER
```

(nota: il flag opzionale "-PSERVER" fa sì che i file di configurazione applicativa con i valori di default presenti nei sorgenti non vengano inseriti nel JAR compilato; sarà pertanto necessario un file di configurazione applicativa esterno al JAR nel momento in cui si lancerà l'applicazione)
5. Nel caso in cui la build avvenga con successo verrà mostrato un messaggio "BUILD SUCCESSFUL" sulla console; il file JAR binario compilato (nome file: mypivot4-be-XXX.jar, dove XXX indica la versione corrente) sarà disponibile nella cartella \$PRJ_ROOT/mypivot4-be/build/libs

Procedura di lancio del server:

1. Posizionarsi nella cartella in cui è presente il file JAR compilato durante la fase di build (mypivot4-be-XXX.jar); nota: se considerato utile, il file in questione può essere spostato in qualsiasi cartella, nel caso si desideri lanciare il server da una cartella differente da quella di build
2. Verificare che nella stessa cartella in cui è presente il JAR sia presente il file di configurazione application.properties con tutte le proprietà di configurazione definite nel paragrafo 1.2 di questo documento;
3. Lanciare il server Spring Boot con il comando:

```
java -jar mypivot4-be-XXX.jar
```

Verifica stato server:

1. Verificare che il processo S.O. del comando Java di avvio del server sia in esecuzione

2. Verificare che sulla console del processo Java e sul file di log applicativo (il path è definito dai parametri di configurazione applicativa) non ci siano errori "fatali"
3. Se il servizio Spring Boot Actuator di monitoraggio applicativo è stato abilitato nella configurazione applicativa (proprietà "management.*"), verificare nella response JSON della chiamata HTTP REST (metodo GET) "/actuator/health" relativa al "context root" su cui è esposta la webapp che tutti i componenti siano in stato "UP"

5.3.2 Installazione front-end (Angular)

Prerequisiti:

1. Node.js v12 (LTS)
2. Un HTTP Server (es. Apache HTTPD o Nginx), correttamente configurato per servire la webapp Angular

Procedura di build:

1. Scaricare i sorgenti di MyPivot (si rimanda alla precedente sezione relativa alla build del back-end)
2. Posizionarsi nella root del progetto "frontend" (path: \$PRJ_ROOT/mypivot4-fe)
3. Aggiornare le dipendenze NPM con il comando:
`npm install`
4. Lanciare la build con il comando:
`npm run build -- mypivot4-fe --base-href <context-root> --output-path <output-path> --configuration production`
dove:
 - a. context-root: é la context-root su cui verrà deployata la webapp
 - b. output-path: path assoluto su cui verrà posizionato il codice "compilato" della webapp
 - c. Il pezzo "--configuration production" è opzionale e attiva una serie di ottimizzazioni sul compilato che ne riducono le dimensioni e ne migliorano le performance (ma rendono il debug più complicato)

Procedura di deploy:

1. Aggiungere nel caso non esista (o verificare che il contenuto sia corretto, nel caso esista) il file \$OUTPUT_PATH/assets/conf/environment.json con il seguente contenuto:

```
{ "baseApiUrl": "<CONTEXT-ROOT-API>", baseMyIntranetUrl: "<ABSOLUTE-URL-MY-INTRANET" }
```
2. Avviare il server HTTP scelto e configurato per servire la webapp MyPivot posizionata nella cartella \$OUTPUT_PATH e sul context-path definito durante la fase di build

Verifica stato server:

1. Aprire con un browser la url che punta al server HTTP scelto e con il context-path di MyPivot; verificare che si apra la homepage di MyPivot4

5.4 INSTALLAZIONE PER DEPLOYMENT IN AMBIENTE KUBERNATES

La struttura della cartella "mypivot4.deploy" conterrà le cartelle, tutte con la stessa struttura di base per ciascun ambiente previsto, contenenti gli artefatti specifici di ogni ambiente "<environment>" di rilascio. Ad es. nella cartella "collaudo" troviamo i template dei descrittori Kubernetes e le configurazioni dei componenti per l'ambiente di collaudo.

Per esemplificare descriviamo di seguito quella dell'ambiente di collaudo.

mypivot4.deploy/base contiene le componenti comuni a tutti gli ambienti:

- /mypivot4-fe: si trovano i descrittori kubernetes

- /mypivot4-be: si trovano i descrittori kubernetes

- /mypivot4-batch: si trovano i descrittori kubernetes

pivot.deploy/overlays/<ambiente> contiene i manifest, i secret e le configurazioni specifiche per ambiente, nelle seguenti sottocartelle:

- /config/<modulo>: si trovano i file di configurazione specifici per l'ambiente e per il modulo

- /env : contiene i secret specifici e per moduli con le relative variabili d'ambiente

- /patches : contiene i manifest delle patch da applicare per l'ambiente

- /pvc : contiene i manifest dei claim da creare per l'ambiente

6 CARICAMENTI INIZIALI

Per quanto riguarda l'impianto iniziale del sistema si possono distinguere due casi:

- Impianto da sistema preesistente in una versione precedente
- Impianto ex novo senza sistema preesistente

6.1 Impianto da sistema preesistente in una versione precedente

nel caso si parta da un sistema preesistente in una versione precedente i passi che devono essere eseguiti sono:

Export dei database

- MyPivot

Eventuale Export del FileSystem relativo ai dati caricati e scaricati e relativi ai flussi
Import degli export precedentemente eseguiti
Eventuale Import dell'export del FileSystem precedentemente eseguito

Applicazione degli script di adeguamento delle banche dati per l'aggiunta di tabelle, attributi, chiavi, indici, disponibili all'interno del repository dei sorgenti nella cartella db del repository precedentemente citato.

Con tale procedura sono garantiti queste macroattività:

- recupero dati da sistema legacy e migrazione nella nuova base dati
 - popolamento massivo di tabelle di codifica
 - profilatura di utenti base per la procedura (admin, lettore, ...)
 - autorizzazione d'accesso al sistema da parte di utenti base
- in quanto saranno preservati dati, utenti, autorizzazioni.

6.2 Impianto ex novo senza sistema preesistente

Nel caso in cui non si dovesse partire da una situazione pregressa sarà disponibile un export con il popolamento delle sole tabelle di codifica e di configurazione e con l'utenza base amministrativa del sistema, in tal caso il dump del database è presente nella cartella: MyPivot -> mypivot.sources -> mypivot-db, sono presenti i file di dump in formato pain, e le tabelle di lookup in formato csv.

7 INSTALLAZIONE JOB

Nei sorgenti le installazioni dei job (i compilati) si possono trovare nella cartella **batch**

Procedura di build

Le procedure per la creazione delle release relative ai job sono due:

- Build tramite Talend
- Build Spring Boot

Build job Talend

La fase di build dei processi prevede la compilazione dei batch tramite applicativo Talend partendo dai sorgenti presenti nella cartella MyPivot.sources :
" MyPivot -> mypivot.sources -> mypivot-batch -> <JOB_TARGET>"

Build job Spring Boot

I batch spring boot sono compilabili utilizzando le procedure descritte nel paragrafo 1.3.1 (Installazione Backend - build)

Procedura di Deploy

La procedura di deploy consiste in:

- copiare i jar generati nella cartella " MyPivot -> mypivot.deploy -> base -> mypivot -batch -> jobs -> batch -> jobs -> <jJOB_TARGET>"
- copiare le librerie generate nella cartella " MyPivot -> mypivot.deploy -> base -> mypivot -batch -> jobs -> batch -> jobs -> <jJOB_TARGET> -> lib"

Piu nel dettaglio:

/batch/bin : contiene gli script per lanciare i job talend

```
<nome del job>_run.sh      per lanciare il job talend
mainJob<nome del job>.sh   lancio del job java da parte del job talend
```

/batch/conf: contiene le configurazioni dei i job talend.

Le sottodirectory hanno il nome del job, e all'interno contengono file di configurazione

/batch/jobs: contiene i job talend compilati (una sottodirectory con i file principali ed una seconda con le librerie. Le sottodirectory hanno il nome del job

/batch/lib: contiene tutte le librerie (.jar) condivise fra i vari job.

A queste librerie di base vanno aggiunte quelle di mypivot aggiornate (mypivot*.jar)

/batch/logs: contiene i log dei job talend. Le sottodirectory hanno il nome del job

/batch/xsd: xsd da usare per alcune validazioni da parte dei job

/batch/report: report jasper da usare nelle stampe

/batch/fonts: font da usare nelle stampe

/batch/images: immagini da usare nelle stampe

I parametri presenti nei file con estensione *.properties, validi per ciascun job nei diversi path della radice **/batch/conf** , sono riconducibili a quelli indicati nella tabella del paragrafo 1.2