



Interreg



UNIONE EUROPEA

MARITTIMO-IT FR-MARITIME



**MOBI
MART**

PROGETTO MOBIMART

T 2.2.1 Set di Api

Maggio- Mai 2020



La cooperazione al cuore del Mediterraneo
La coopération au cœur de la Méditerranée

Progetto numero	168
Acronimo	MOBIMART
Titolo progetto	Mobilità intelligente mare terra
Inizio/ fine progetto	01.04.2018- 31.03.2021
Durata	36 mesi

Componente	T 2
Attività a cui si riferisce il prodotto	Sviluppo Application Programming Interfaces (API)
Titolo prodotto	Test di funzionamento della Piattaforma Cloud.
Stage di riferimento	P4
Periodo riferimento (inizio/Fine)	1/09/2019 - 31/03/2020

Autore	Silvia Antonelli, Alexandr Tardo, Domenico Lattuca
Versione	2019-2 (secondo piano di sviluppo)
Data	31/5/2019
Responsabile della validazione	Paolo Pagano
Data revisione	
Riepilogo modifiche	
Autore modifiche	

Storia Documento

Versione	Data	Annotazioni
1.0	21/01/2020	Realizzazione della struttura del documento
1.1	27/01/2020	Scrittura della procedura di testing
1.1	05/02/2020	Completamento deliverable, prima revisione
1.2	10/02/2020	Seconda revisione
1.3	27/05/2020	Aggiornamento paragrafo “Implementazione delle API”

Bibliografia

Ref.	Oggetto	Link
1	Report Task Force 17 TM2.0 – MaaS (draft)	-
2	PMIS	http://www.guardiacostiera.gov.it/mezzi-e-tecnologie/Pages/PMIS.aspx

Indice/Index

Introduzione	p. 2
La piattaforma “multi-porto”: l’integrazione dei porti di Cavo, Portoferraio e Capraia Isola	p.4
Implementazione delle API	p.5

1. Introduzione

Fin dal momento della sua realizzazione, la piattaforma MOBIMART¹ ha subito un processo di continua e rapida evoluzione. L'idea degli sviluppatori è sempre stata infatti quella di creare un punto di raccolta dati, relativo ai principali porti della Toscana, capace di mutare adattandosi alle principali tendenze di mercato.

Partendo da questa premessa, una volta definiti e testati i servizi offerti dalla piattaforma MOBIMART per il Porto di Livorno, lo step successivo è stato proprio quello di ampliare l'offerta aggiungendo l'insieme dei porti della Toscana che, insieme al già citato porto di Livorno, costituiscono la spina dorsale del traffico merci e passeggeri regionale.

Il processo di evoluzione della piattaforma non si è però fermato all'aggiunta dei nuovi porti succitati. Sono state infatti implementate due API: servizi atti a potenziare le funzionalità di MOBIMART che adesso è in grado di offrire nuovi metodi di download automatico, ad esempio, ad uno sviluppatore interessato alla lista delle navi in partenza da un determinato porto.

In aggiunta, al contrario della precedente versione della piattaforma, che riusciva a visualizzare solo ed esclusivamente le navi in partenza dal porto di Livorno, quella attuale è in grado di fornire l'elenco aggiornato sia delle navi in partenza da un determinato porto, selezionato dall'utente, che delle navi in arrivo presso un determinato porto selezionato anche in questo caso dall'utente. Un ulteriore upgrade riguarda infine l'inserimento di nuovi file GTFS specifici per i porti di Piombino e Portoferraio (la Figura 1 presenta una schermata della piattaforma).

La scelta del software di testing è ricaduta su uno strumento completamente open-source come Static GTFS Manager capace di importare un file GTFS e mostrarne il contenuto ricreando automaticamente le varie tabelle che compongono il file inserito, generarne le dipendenze e mostrarne il contenuto (visualizzando su mappa sia i vari porti che le differenti rotte avvalendosi dell'ausilio dei file contenuto proprio all'interno dell'archivio GTFS). Ciò rappresenta dunque un ottimo strumento capace di misurare la qualità dei dati generati dalla piattaforma.A

Infine, il risultato dei test si è dimostrato perfettamente in linea con le nostre aspettative. Ciò non esclude in futuro ulteriori modifiche volte a migliorare ulteriormente la qualità dei dati a nostra disposizione o ad ampliarne la quantità.

¹ Sito internet: <https://mobimarttimetable.labtclivorno.it>

GTFS Livorno GTFS Piombino GTFS Portoferraio

LIVORNO PARTENZE Data Search:

Show 10 entries

DATA	ORARIO	COMPAGNIA	NOME NAVE	PORTO DI DESTINAZIONE/ARRIVO	COD. PORTO	BANCHINA
27/05/2020	04:45	GRIMALDI LINES	EUROCARGO CAGLIARI	SAVONA	ITSVN	24/SUD
27/05/2020	08:30	TOREMAR	LIBURNA	CAPRAIA	ITCPA	66/NORD/R
27/05/2020	10:30	GRIMALDI LINES	EUROCARGO BARI	SALERNO	ITSAL	24/NORD
27/05/2020	11:00	GRIMALDI LINES	CRUISE BONARIA	OLBIA	ITOLB	14/F/R
27/05/2020	13:30	SARDINIA FERRIES	PASCAL LOTA	BASTIA	FRBIA	62/R
27/05/2020	13:30	SARDINIA FERRIES	MEGA EXPRESS	BASTIA	FRBIA	62/R
27/05/2020	16:30	TOREMAR	LIBURNA	CAPRAIA	ITCPA	66/NORD/R
27/05/2020	20:00	TIRRENIA	BENIAMINO CARNEVALE	CAGLIARI	ITCAG	20/R
27/05/2020	21:00	SARDINIA FERRIES	CORSICA MARINA SECONDA	GOLFO ARANCI	ITGAI	64/NORD/R
27/05/2020	21:30	GRIMALDI LINES	EUROCARGO PALERMO	SAVONA	ITSVN	24/NORD

Showing 1 to 10 of 39 entries 1 2 3 4



Figura 1 - Piattaforma Mobimart Aggiornata

2. La piattaforma “multi-porto”: l’integrazione dei porti di Cavo, Portoferraio e Capraia Isola

La prima versione della piattaforma MOBIMART prevedeva che tutti i servizi offerti riguardassero solo ed esclusivamente il Porto di Livorno, in qualità di centro nevralgico e operativo del traffico portuale della Regione Toscana. Da questa scelta è scaturita in primis la realizzazione di un tabellone orario contenente diverse tipologie di informazioni riguardanti le navi in partenza dal porto di Livorno (ad es. data, orario, compagnia, banchina), e in secondo luogo la realizzazione di un file General Transit Feed Specification (GTFS) scaricabile da un utente interessato a conoscere i dati relativi alle navi che transitano dal porto oggetto dell’attuale discussione.

Quanto è stato appena descritto rappresenta solo il primo step della realizzazione della piattaforma MOBIMART, completato il quale è stata avviata una fase di testing, conclusasi con successo e della quale saranno riepilogati più avanti i risultati. Il passo successivo riguardante l’estensione dei servizi, già presenti e attivati per il Porto di Livorno, ai rimanenti porti della Toscana si è dunque rivelato quasi fisiologico.

Si fornisce di seguito l’elenco dei porti integrati, accompagnato da una breve descrizione di per ognuno di loro:

Porto di Piombino: si tratta di un porto polifunzionale, dedicato principalmente al traffico passeggeri con l’Isola d’Elba, la Sardegna e con la Corsica (stagionalmente) e al traffico merci, per la maggior parte rinfuse, destinate ai mercati nazionali ed internazionali;

Porto di Portoferraio: è il principale approdo dell’Isola d’Elba, situato sulla costa settentrionale in posizione ben protetta all’interno di un’ampia baia. Il suo utilizzo è strettamente legato al traffico traghetti di linea con Piombino, al traffico crociere e al traffico da diporto;

Porto di Rio Marina: in passato ha acquisito notevole importanza grazie all’esportazione del minerale di ferro estratto nelle cave che dominano l’omonimo abitato. Oggi lo scalo consiste sostanzialmente in un terminal per le navi traghetto, in collegamento con il porto di Piombino, le quali trovano ormeggio nell’ultima parte della diga (lunga circa 80 m);

Pontile di Cavo: è un porto turistico, con alcune funzioni commerciali, che ospita l’accosto per gli aliscafi di linea destinati alla rotta Piombino-Cavo-Portoferraio e nel periodo estivo l’accosto delle navi traghetto nel servizio di linea Piombino-Cavo;

Porto di Capraia Isola: Capraia è per estensione la terza isola dell’Arcipelago Toscano con una superficie di 19,3 kmq. In virtù dell’elevata distanza dalla costa continentale situata ad est (ben 54 km) è una tra le isole più distanti dalla penisola italiana.

Terminata questa breve introduzione riguardante i porti che sono quindi entrati a far parte della piattaforma MOBIMART, si procede alla descrizione delle varie fasi di implementazione dei nuovi servizi.

3. Implementazione delle API

Le API (acronimo di Application Programming Interface) sono set di definizioni e protocolli attraverso cui vengono realizzati e integrati software applicativi. Le API consentono la comunicazione tra servizi differenti senza che vi sia la necessità di sapere come vengano implementati tali servizi, semplificando in questo modo lo sviluppo dei vari applicativi e consentendo un netto risparmio di tempo e denaro.

Durante la creazione di nuovi strumenti e prodotti o durante la gestione di quelli esistenti, le API offrono flessibilità, semplificano la progettazione, l'amministrazione e l'utilizzo e garantiscono opportunità di innovazione.

Nel caso della piattaforma MOBIMART, la realizzazione delle API descritte a breve è stata resa possibile dall'utilizzo delle OpenAPI Specification (in passato "Swagger Specification"), ovvero un formato di descrizione per le API di tipo REST (acronimo di "REpresentational State Transfer) che indica una rappresentazione del trasferimento di stato di un determinato dato.

Il processo di implementazione di un'API consiste nella creazione di una serie di endpoint (semplici URL) ai quali corrispondono determinate operazioni che vengono eseguite in base alle richieste provenienti da un programma esterno, permettendo così di rispondere alle richieste fatte da uno sviluppatore.

Un file scritto seguendo le specifiche "OpenAPI" consente di descrivere l'intera API permettendo la descrizione di:

Endpoint disponibili (/ utenti) e le operazioni possibili su ciascuno di questi endpoint (GET / utenti, POST / utenti);

Parametri operativi in ingresso e in uscita per ogni operazione;

Metodi di autenticazione;

Informazioni di contatto, licenza, condizioni d'uso e/o altre informazioni.

È possibile scrivere le specifiche sia in formato YAML che in formato JSON: entrambi facili da imparare e altamente leggibili sia per l'uomo che per la macchina. Una volta definita la specifica OpenAPI e gli strumenti Swagger (utili nella progettazione, costruzione e documentazione delle API REST), lo sviluppo può articolarsi in modi differenti.

Nel caso della piattaforma MOBIMART, la scelta è stata quella di scrivere le API utilizzando un formato YAML come riferimento. Gli endpoint realizzati e messi a disposizione dell'utente che vuole sfruttare le API previste dal progetto sono i seguenti:

Endpoint /departures/{port}: quando interrogato, fornisce come risultato l'elenco delle navi in partenza da un porto, passato come parametro, a partire da una data inserita dall'utente (o dalla data odierna qualora non venisse fornita alcuna data) fino all'ultimo giorno utile presente nel previsionale;

Endpoint /gtfs/{port}: quando interrogato, fornisce come risultato il file GTFS (codificato in JSON) relativo alle navi in partenza da un porto e in arrivo in un altro porto, entrambi passati come parametri, nel periodo specificato dall'utente (o dalla data odierna fino all'ultimo giorno utile presente nel previsionale, qualora non venisse specificato alcun periodo).

In Figura 2 è possibile osservare la descrizione dettagliata dei due endpoint appena menzionati, oltre ai loro parametri e alle risposte che ritornano nel caso in cui l'interrogazione si concluda con successo o meno.

```

/departures:
  get:
    description: 'Returns the list of departures from port specified in {port} and starting from current date-time or from that (eventually) specified in the query'
    parameters:
      - $ref: '#/components/parameters/PortName'
      - $ref: '#/components/parameters/DeparturesLimit'
      - $ref: '#/components/parameters/StartingDepartureDate'
      - $ref: '#/components/parameters/ArrivalPortName'
    responses:
      '200':
        description: Successfully return a list of departures
        content:
          application/json:
            schema:
              type: array
              items:
                $ref: '#/components/schemas/Departure'
      '400':
        $ref: '#/components/responses/400Error'

/gtfs:
  get:
    description: "Returns a set of GTFS encoded in JSON syntax"
    parameters:
      - $ref: '#/components/parameters/PortName'
    responses:
      '200':
        description: Successfully return a set of JSON GTFS
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/GTFS'
      '400':
        $ref: '#/components/responses/400Error'

```

Figura 2 - Struttura degli endpoint

In Figura 2 - Struttura degli endpoint sono invece mostrate le caratteristiche che devono avere i parametri “passati” dall’utente oltre allo schema dei due endpoint, cioè la tipologia dei vari campi che li compongono e l’indicazione relativa ai campi obbligatori.

Entrando nel dettaglio, i parametri richiesti sono solamente cinque e qui di seguito vengono elencati:

- PortName: rappresenta il nome del porto di partenza;
- StartingDepartureDate: data iniziale inserita dall’utente e di riferimento per la query;
- DepartureLimit: data finale di riferimento per la query nel caso in cui non sia specificata un data di inizio (se tale data è successiva all’ultima data utile presente nel previsionale, il risultato della query terrà conto solo di quest’ultima);
- ArrivalPortName: rappresenta il nome del porto di arrivo.

Per quanto riguarda invece lo schema degli endpoint è immediato osservare che, pur essendo questi ultimi rappresentati entrambi dallo stesso tipo object, le proprietà sono sensibilmente differenti.

L’endpoint per i file GTFS prevede solo ed esclusivamente l’inserimento di un parametro, in modo che l’utente possa specificare quale sia il porto di riferimento (nello schema in **Errore. L’origine riferimento non è stata trovata.** denominato PortName) per il quale si richiede il file GTFS (ad

es. Livorno, Piombino o Portoferraio). Una volta interpellato, l'endpoint assegnerà ad ogni campo un semplice riferimento relativo al file di testo localizzato all'interno del relativo archivio GTFS2.

² La struttura di un file GTFS (*General Transit Feed Specification*) è stata dettagliatamente discussa nel precedente Deliverable 3.3, relativo alla realizzazione delle interfacce di interscambio e interoperabilità.