

# VPCS Legacy

## Integrazione web services

## Sommario

Sommario .....	2
1 Introduzione .....	3
1.1 Web services .....	3
1.1.1 Gestione credenziali per accesso .....	3
1.1.2 Specifiche generali per la fruizione .....	3
1.1.3 SendBaplieStatus .....	8
1.1.4 SendWithdrawBlock .....	9
1.1.5 SendLandingOperation .....	10
1.1.6 SendMeansOfTransportLoaded .....	11
1.1.7 SendTerminalGateOutOperation .....	13
1.1.8 SendMma .....	14
1.1.9 SendMmaResult .....	15
1.1.10 SendDu .....	16
1.1.11 SendLandingFile .....	16
1.1.12 SendTerminalGateOutFile .....	17

# 1 Introduzione

Il presente documento riepiloga gli interventi apportati al VPCS Legacy per le attività web services.

## 1.1 Web services

Nella web app del VPCS Legacy sono stati introdotti dei web services mediante **WebMethod**.

### 1.1.1 Gestione credenziali per accesso

#### 1.1.1.1 Profilo utente

Per ciascun utente, dalla pagina Profilo utente è stata introdotta la nuova tab *Interazione web service* in cui può accedere alle credenziali da utilizzare per interfacciarsi con i web services:



Token utente:	VPCS-1740373031
Chiave per TOTP:	tz37IVsS2
Abilitato:	<input checked="" type="checkbox"/>

Figura 1 - Token e chiave per TOTP in profilo utente

Da questa interfaccia l'utente ha quindi evidenza se per il proprio profilo è disponibile un token (in caso contrario può chiederne tramite helpdesk), e se lo stesso è abilitato.



Interoperabilità web service non abilitata: contattare Help Desk.

Figura 2 - Profilo senza credenziali WS

#### 1.1.2 Specifiche generali per la fruizione

I web services sono finalizzati alla ricezione di dati per il VPCS Legacy, per tanto sono implementati come **Web Method** ed esposti direttamente dalla web application del VPCS.

I web services sono accessibili tramite metodo **POST** e *Content-Type application/json; charset=UTF-8* e su connessione SSL. La struttura dei messaggi in ingresso e uscita dei web services seguono lo standard JSON.

#### 1.1.2.1 Autenticazione

Nelle attuali logiche di business relative gli upload file da interfaccia, il VPCS Legacy fa ampio uso di informazioni associate al profilo utente che ha effettuato la login: per gestire il flusso di acquisizione tramite web services è quindi essenziale che sia mantenuto il riferimento all'utente che sta effettuando l'operazione.

Anziché gestire la login via web service tramite stesse *username* e *password* impiegate per la login classica (sensibili di blocco e scadenza password), si è implementata l'associazione all'utente di due nuove chiavi che dovranno essere utilizzate per l'accesso di due nuovi valori associati ad un profilo utente:

- **token utente univoco**: identificativo utente per l'accesso al web service; ○ pensato per non essere modificato dopo l'assegnazione (salvo casi particolari);
- **codice TOTP** generato a partire da una chiave nota (e dedicata) a ciascun utente; ○ pensato per essere aggiornato su richiesta del sistema del terminal (modifica che va fatta contemporaneamente lato VPCS e lato TOS).

Ciascun utente può avere accesso a *token utente* e *chiave per TOTP* dal proprio profilo utente.

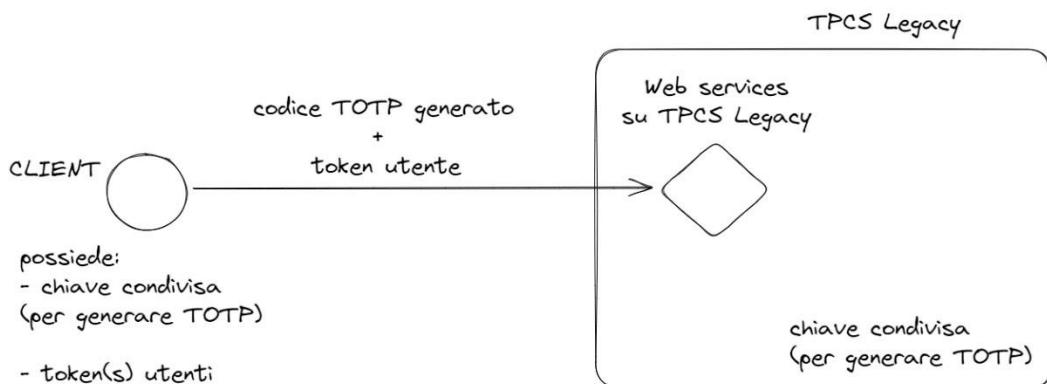


Figura 3 - Autenticazione nuovi WS su VPCS Legacy

A tutti i web services va fornito un parametro **VPCSPayload** in formato JSON, di base strutturato come segue:

Struttura di VPCSPayload	Tipo	Descrizione
credentials	totpCode	string Codice TOTP, generato dalla chiave condivisa. <u>Per generare il codice: Step: 30, SHA-256</u>
	usrToken	string Token utente
data		Contiene le proprietà specifiche del servizio invocato

Nella sezione **credentials** vanno indicate le informazioni per consentire l'effettiva fruizione dei servizi:

- **usrToken**: token dell'utente associato al profilo VPCS Legacy che intende effettuare l'operazione;
- **totpCode**: codice TOTP generato su chiave condivisa

Ipotizzando che la chiave condivisa sia **pRovA1203** (totpSecret), per generare il **totpCode** il client deve:

1. ottenere il *base32* della chiave **pRovA1203** (ovvero, **OBJG65SBGEZDAMY=**)
2. passare il codice in *base32* alla funzione che genera il codice Totp, impostando **Step: 30 e SHA-256**

I linguaggi di programmazione predispongono librerie dedicate per la corretta generazione del totpCode, per esempio in .NET esiste **OtpNet.Totp**.

```
new OtpNet.Totp(Encoding.ASCII.GetBytes(totpSecret), 30, OtpHashMode.Sha256);
```

Il valore così ottenuto sarà un numero, per esempio **637638**, e dovrà essere fornito quindi al campo `totpCode`

### Esempio

```
{
  "VPCSPayload": {
    "credentials": {
      "usrToken": "VPCS-0174025036",
      "totpCode": "637638"
    },
    "data": {
      ...
    }
  }
}
```

**Importante:** assicurarsi di generare il codice TOTP e assegnarlo al payload subito prima di effettuare la chiamata al web service di VPCS, per ogni invocazione.

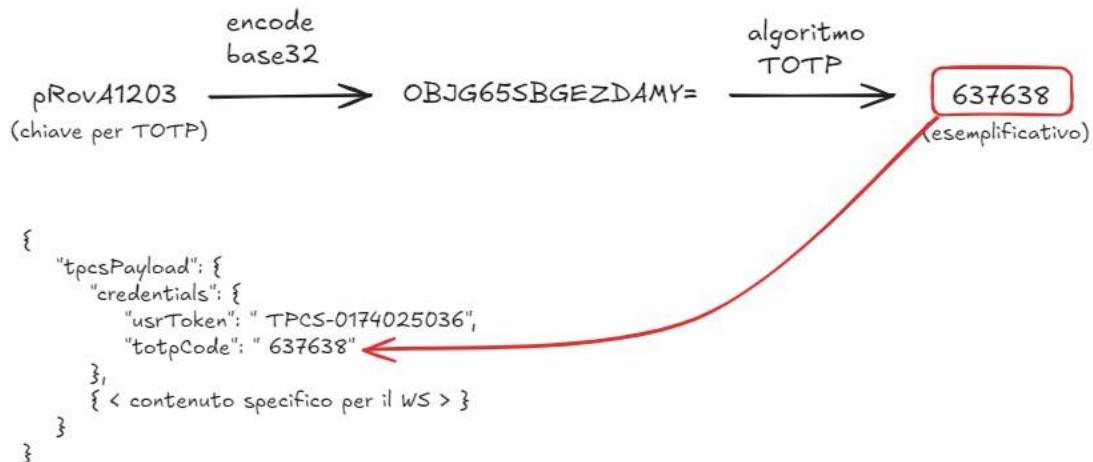


Figura 4 - generazione `totpCode`

La fruizione dei web services è permessa nel rispetto delle seguenti condizioni:

- il token utente è presente e attivo su VPCS;
- il codice totp fornito è valido;
- il profilo utente associato al token può utilizzare il servizio.

In caso contrario, il messaggio trasmesso non verrà gestito e al mittente verrà restituito un errore con un messaggio informativo: per utilizzare i servizi l'utente dovrà effettuare le operazioni indicate nel messaggio, accedendo al VPCS o chiedendo il supporto dell'assistenza.

Oltre alla sezione **credentials** saranno presenti altre proprietà, in genere nel campo **data**, caratterizzate dal web service specifico.

### 1.1.2.2 Risposta dei web services

**IMPORTANTE:** Web Method prevede che l'effettivo contenuto della risposta venga incapsulato a sua volta in un JSON costituito dalla proprietà **d**:

- { **d**: { json della risposta } }

Pertanto, per accedere al JSON effettivo della risposta, il chiamante deve prima accedere alla proprietà **d**.

Tutti i web services risponderanno con un JSON costituito dalle seguenti proprietà:

Proprietà	Tipo	Descrizione
requestStatus	string	Feedback esito dell'operazione (OK, OK_WITH_WARNING, ERRORS, UNEXPECTED_ERRORS)
message	string	valorizzato per fornire dettagli testuali al chiamante: al verificarsi di errori, fornisce informazioni sulla causa.
warningMessage	string	valorizzato al verificarsi di situazioni che richiedono attenzione (es. il login all'applicativo per accertarsi di quanto trasmesso)
codeMessage	string	codice univoco che identifica una particolare categoria di messaggi. Sarà vuoto se nessun messaggio presente o non codificato
dataRes	object	Eventuali dati specifici, risposta del web service invocato. <u>Se non utilizzato, viene valorizzato a null.</u>

I valori di **requestStatus** previsti saranno i seguenti:

- **OK**: la richiesta è stata gestita correttamente;
- **OK\_WITH\_WARNING**: la richiesta è stata gestita correttamente, ma ha generato un avviso per l'utente il cui dettaglio verrà riportato in **warningMessage**. Al verificarsi di un warning, consigliamo all'utente di accedere all'applicativo VPCS personalmente e verificare quanto è stato notificato nel messaggio ricevuto o Un esempio di warning è *password in scadenza*;
- **ERRORS**: la richiesta non è stata soddisfatta, la causale verrà riportata nella proprietà **message** e codificata in **codeMessage** se errore gestito;
- **UNEXPECTED\_ERRORS**: la richiesta non è stata soddisfatta per il verificarsi di un errore non atteso.

I valori di **codeMessage** dipenderanno dalla natura di ciascun web service e dai dati gestiti. Di seguito quelli generici che si prevede di introdurre:

- **CREDENTIALS\_ERROR**: le credenziali fornite non sono corrette;
- **NOT\_ALLOWED**: le credenziali sono associate ad un utente che non può utilizzare il servizio (non approvato, token scaduto...);
- **MISSING\_PARAMETERS**: nella richiesta non tutti parametri sono valorizzati per consentire una corretta gestione;

Esempio di risposta per **trasmissione senza errori**:

```
{  
  
    "d": {  
        "requestStatus": "OK",  
        "message": "",  
        "warningMessage": "",  
        "codeMessage": "",  
        "dataRes": null  
    }  
}
```

Esempio di risposta per **trasmissione senza errori ma con warning**:

```
{  
  
    "d": {  
        "requestStatus": "OK_WITH_WARNING",  
        "message": "",  
        "warningMessage": "E' in uso un tracciato obsoleto",  
        "codeMessage": "",  
        "dataRes": null  
    }  
}
```

Esempio di risposta per **trasmissione con errori gestiti** (relativi al contenuto del file o dati su sistema VPCS):

```
{  
  
    "d": {  
        "requestStatus": "ERRORS",  
        "message": "Riga 2 - Valore non previsto al campo 10",  
        "warningMessage": "",  
        "codeMessage": "DU_DETAIL_ERROR",  
        "dataRes": null  
    }  
}
```

Esempio di risposta per **trasmissione con errori non gestiti**:

```
{  
  
    "d": {  
        "requestStatus": "UNEXPECTED_ERRORS",  
        "message": "Errore non gestito",  
        "warningMessage": "",  
        "codeMessage": "",  
        "dataRes": null  
    }  
}
```

#### 1.1.2.3 *Limiti infrastrutturali*

I web services potranno presentare una latenza di risposta strettamente legata alla dimensione del JSON comunicato: in particolare, per quei servizi che ricevono in ingresso file codificati in base64, inciderà la dimensione di quest'ultimo e le procedure di verifica intrinseche del servizio.

Inizialmente verrà impostata una dimensione massima della lunghezza del JSON a **50 MB**, fermo restando che la dimensione massima supportata è di 2048 MB.

#### 1.1.2.4 *Convenzioni per la fruizione*

Si chiede all'operatore di implementare il proprio sistema client al fine di adottare le seguenti convenzioni:

- invocare lo stesso servizio con parsimonia e in modo cadenzato (attendere dai 2 ai 5 secondi tra un'invocazione e l'altra), questo per limitare situazioni di sovraccarico dei servizi esposti (in particolare quelli che gestiscono file);
- prestare attenzione alle risposte fornite da VPCS e ai messaggi generali, in particolare
  - quando **requestStatus** è `OK_WITH_WARNING` o `ERRORS` attraverso la proprietà `codeMessage` è possibile intercettare un particolare categoria dei messaggi
  - quando **requestStatus** è `UNEXPECTED_ERRORS` ;
- in caso di variazioni ai tracciati, regole di composizione del nome file o alla struttura intera del JSON, gli operatori, informati per tempo, dovranno adeguare i loro sistemi per continuare a colloquiare correttamente con VPCS.

Nell'utilizzo dei web services possono verificarsi situazioni in cui la risposta restituita è una pagina HTML anziché un `json`: ciò può avvenire quando non si invoca correttamente il servizio (URL errata o JSON di input non rispetta la struttura prevista), oppure se si verifica un errore interno al server. In questo caso, l'operatore dovrà contattare HelpDesk ed essere in grado di fornire un file con estensione `.json` l'intero contenuto della richiesta trasmessa.

### 1.1.3 *SendBaplieStatus*

Serve per automatizzare il processo senza che il terminal debba accedere al VPCS per eseguire l'operazione (si ricorda che l'apposizione del baplie ad una sezione di manifesto di arrivo serve per abilitare la trasmissione delle distinte di uscita verso il sistema del terminal, e agisce a livello di sezione MMA per quel terminal). Il servizio applica il baplie solo alle sezioni manifesto che corrisponde ai codici magazzino comunicati (il sistema verificherà la corrispondenza con il codice magazzino).

Il profilo utente che può utilizzare questo web service è: **terminalista**.

Il messaggio da passare al web service ha questa struttura:

Struttura di VPCSPayload		Tipo	Descrizione
credentials			
data	manifestCode	string(18)	Codice manifesto di arrivo [O]
	warehouseCodes	string[]	Sequenza dei codici magazzino per individuare la sezione eMMA di riferimento a cui applicare il baplie [O]

In caso di errore il web service può ritornare uno dei possibili codici:

- **CREDENTIAL\_ERROS**: credenziali fornite non riconosciute;
- **NOT\_ALLOWED**: l'utente non è autorizzato ad usare il servizio;
- **MMA\_NOT\_FOUND**: non è stato individuato il manifesto di riferimento;
- **MMA\_SECTION\_NOT\_FOUNT**: non è stata individuata una sezione manifesto di riferimento;

#### 1.1.4 SendWithdrawBlock

Il Terminal per i rotabili manifesta un problema legato alla conformità del sigillo: nell'eManifest il riferimento al sigillo per i rotabili viene inserito nel campo “marca dei colli”, insieme alla targa del mezzo. Se il sigillo non è conforme, il Terminal non può far uscire il mezzo, fino alla verifica di conformità da parte della Dogana.

Il servizio consente al terminal di comunicare una generica difformità e di impostare l'impedimento al ritiro.

**Nel caso della difformità inerente il sigillo**, il flusso sarà il seguente:

1. Invocazione di questo servizio per bloccare il ritiro (se importante per il terminal);
2. creazione di una pratica ispettiva nel modulo controllo a piazzale; alla conclusione dell'ispezione e alla ricezione della chiusura operativa dal terminal verrà ricevuto il corretto codice sigillo;
3. Invocazione di questo servizio per sbloccare il ritiro.

Si precisa che

- l'utilizzo di questo web service non è obbligatorio e dipende dal terminal;
- è alternativo all'utilizzo dell'impedimento al ritiro tramite DUB. Perciò se impostato tramite web service, alla ricezione di una DUA dovrà essere mantenuto in risposta alla DUB.

Il profilo utente che può utilizzare questo web service è: **terminalista**.

Il messaggio da passare al web service ha questa struttura:

Struttura di VPCSPayload	Tipo	Descrizione
credentials		
data	manifestCode	string(18) Codice manifesto di arrivo [O]
	containerCode	string(11) Codice contenitore di riferimento ([C] in alternativa a plateCode, altrimenti null)
	plateCode	string(40) Codice contenitore di riferimento ([C] in alternativa a containerCode, altrimenti null)
	message	string(100) Indicare causale difformità
	withdrawBlock	boolean Valorizzare con true se riscontrate difformità (blocca il ritiro) Valorizzare con false, per revocare la difformità (sblocca il ritiro)

In caso di errore il web service può ritornare uno dei possibili codici:

- **CREDENTIAL\_ERROS**: credenziali fornite non riconosciute;
- **NOT\_ALLOWED**: l'utente non è autorizzato ad usare il servizio;
- **MMA\_NOT\_FOUND**: non è stato individuato il manifesto di riferimento;
- **UNIT\_NOT\_FOUND**: container / rotabile non trovato;

### 1.1.5 SendLandingOperation

Web service dedicato alla ricezione **dell'informazione di sbarco dei telai**. Le informazioni devono essere organizzate in modo da elencare i telai sbarcati in un certo momento e da una certa nave.

Il profilo utente che può utilizzare questo web service è: **terminalista**.

Il messaggio da passare al web service ha questa struttura:

Struttura di VPCSPayload		Tipo	Descrizione
credentials			
data	header	headerType	Dati nave e manifesto di riferimento
	equipmentChassis	equipmentChassisType[]	Elenco di informazioni telai
Struttura di headerType		Tipo	Descrizione
shipIMO	string(7)	IMO nave [O]	
shipName	string(35)	Nome nave [F]	
voyage	string(256)	Viaggio [F]	
manifestCode	string(18)	Codice manifesto (MRN) [O]	
Struttura di equipmentChassisType		Tipo	Descrizione
identification	string(70)	Codice telaio [O]	
movementDate	string(12)	Data di sbarco [O] yyyyMMddHHmm	
digitalId	string(200)	Id digitale di riferimento [O]	
grossWeightKg	number(10,6)	Peso in kilogrammi [F]	
cancel	boolean	Valorizzare true se cancellazione di un precedente invio [C]	

In caso di errore il web service può ritornare uno dei possibili codici:

- **CREDENTIAL\_ERROS**: credenziali fornite non riconosciute;
- **NOT\_ALLOWED**: l'utente non è autorizzato ad usare il servizio;
- **MISSING\_PARAMETERS**: la richiesta non è valida per non conformità dei parametri in ingresso;

- **MMA\_NOT\_FOUND**: non è stato individuato il manifesto di riferimento;
- **MISSING\_IDENTIFICATION**: la richiesta non è valida: uno degli identificativi telai comunicati non è presente a sistema.

### 1.1.6 SendMeansOfTransportLoaded

Per ricevere in VPCS, prima dell'uscita dal terminal

- i telai caricati su una data bisarca (campo **loadedVehicles**)
- oppure, i quantitativi di merce varia/rinfusa caricati su un dato mezzo (campo **loadedGoods**), sia in termini di peso lordo in KG che in numero di colli/pezzi
- oppure, la lista dei contenitori trasportati dal mezzo (campo **loadedContainers**)

Il profilo utente che può utilizzare questo web service è: **terminalista**.

Il messaggio da passare al web service ha questa struttura:

Struttura di VPCSPayload	Tipo	Descrizione
credentials		
data	plateCode	string(35) Targa del mezzo trasmesso (motrice) [O] <b>E' la targa che poi GTS3 invierà al VPCS per i dati doganali</b>
	plateCountryCode	string(3) Nazione del rimorchio / bisarca con il carico [F]
	loadedVehicles	loadedVehiclesType[] Elenco dei veicoli caricati ([C], se telai /auto)
	loadedGoods	loadedGoodsType[] Elenco della merce caricata ([C], se merce varia/sfusa)
	loadedContainers	loadedContainerType[] Elenco container ([C], se merce su container)

Segue la definizione del tipo **loadedVehiclesType** (auto su bisarca)

Struttura di loadedVehiclesType	Tipo	Descrizione
identification	string(70)	Codice telaio [O]
grossWeightKg	number(10,6)	Peso in kilogrammi [F]
receiverVatNumber	string(20)	P.iva ricevitore [F]
consigneeVatNumber	string(20)	P.iva destinatario [F]
mrn	string(18)	Codice MRN della dichiarazione [O]
customsNumber	string(8)	Numero Doganale della dichiarazione [F]

customsNumberCin	string(1)	Numero Doganale CIN della dichiarazione [F]
isLastStock	boolean	Valorizzare true se si tratta dell'ultima giacenza caricata (esaurisce il peso nella dichiarazione), false altrimenti [O]
cancel	boolean	Valorizzare true se cancellazione di un precedente invio [C]

Segue la definizione del tipo **loadedGoodsType** (merce varia/rinfuse)

Struttura di loadedGoodsType	Tipo	Descrizione
manifestCode	string(18)	Codice manifesto di arrivo di riferimento [C]. Non valorizzare se da deposito (H2).
isFromStorage	boolean	Valorizzare true se da deposito (da H2) [C]
grossWeightKg	number(10,6)	Peso in kilogrammi [O]
mrn	string(18)	Codice MRN della dichiarazione [O]
customsNumber	string(8)	Numero Doganale della dichiarazione [F]
customsNumberCin	string(1)	Numero Doganale CIN della dichiarazione [F]
receiverVatNumber	string(20)	P.iva ricevitore [F]
consigneeVatNumber	string(20)	P.iva destinatario [F]
isLastStock	boolean	Valorizzare true se si tratta dell'ultima giacenza caricata (esaurisce il peso nella dichiarazione), false altrimenti [O]
cancel	boolean	Valorizzare true se cancellazione di un precedente invio [C]
packsNumber	number	Numero colli / pezzi [O] (indicare 0 se rinfuse)

Segue la definizione del tipo **loadedContainerType** (container)

Struttura di loadedContainerType	Tipo	Descrizione
containerCode	string(11)	Codice del contenitore. [O]
full	char(1)	Indica se è pieno (Y) o vuoto (N). [O]
operationStatus	string(8)	ADD, UPDATE, DELETE. [O]
operationDate	datetime	Data e ora dell'operazione di caricamento, se disponibile [O] Formato: yyyy-MM-ddThh:mm o yyyyMMddHHmm
isoType	string(4)	Codice ISO del container [O]

seals	Array of string(40)	Elenco di sigilli APPOSTI al container [F]
emptyReEntryTerminalCode	string(3)	Codice terminal noto a VPCS, dove rientrerà il vuoto [F]
shippingCompany	string(400)	Descrizione della compagnia di navigazione [F]
serviceName	string(400)	Servizio (servizio svolto dalla compagnia di navigazione) [F]

In caso di errore il web service può ritornare uno dei possibili codici:

- **CREDENTIAL\_ERROS**: credenziali fornite non riconosciute; ○ **NOT\_ALLOWED**: l'utente non è autorizzato ad usare il servizio;
- **MISSING\_PARAMETERS**: la richiesta non è valida per non conformità dei parametri in ingresso;
- **MISSING\_GOODS\_AND\_VEHICLES**: loadedVehicles e loadedGoods entrambi non valorizzati;
- **MISSING\_VEHICLES**: non è stato individuato l'MRN di riferimento; ○ **MISSING\_MRN**: non è stato individuato l'MRN di riferimento.

### 1.1.7 SendTerminalGateOutOperation

Per ricevere in VPCS l'informazione di **uscita** dal terminal

- dei telati caricati su una bisarca. Si assume che stiano uscendo tutti i telai precedentemente associati alla bisarca;
- oppure, di un mezzo caricato con merce varia/rinfusa; si assume che stia uscendo tutta la merce dichiarata su questa targa.

Il profilo utente che può utilizzare questo web service è: **terminalista**.

Il messaggio da passare al web service ha questa struttura:

Struttura di VPCSPayload	Tipo	Descrizione
credentials		
data	header	headerType Dati nave e manifesto di riferimento
	meansOfTransport	meansOfTransportType Informazioni di uscita bisarca
	cancel	Valorizzare true se cancellazione di un precedente invio [C]
Struttura di headerType	Tipo	Descrizione
shipIMO	string(7)	Imo nave [C] Non valorizzare se da deposito (H2).
shipName	string(35)	Nome nave [F]
voyage	string(256)	Viaggio [F]
manifestCode	string(18)	Codice manifesto (MRN) [C] Non valorizzare se da deposito (H2).
isFromStorage	boolean	Valorizzare true se da deposito (da H2) [C]

Struttura meansOfTransportType	di Tipo	Descrizione
plateCode	string(35)	Targa del mezzo trasmesso precedentemente (motrice) [O]
plateCountryCode	string(3)	Nazione del rimorchio / bisarca con il carico [F]
movementDate	string(12)	Data di uscita [O] yyyyMMddHHmm

In caso di errore il web service può ritornare uno dei possibili codici:

- **CREDENTIAL\_ERROS**: credenziali fornite non riconosciute;
- **NOT\_ALLOWED**: l'utente non è autorizzato ad usare il servizio;
- **MISSING\_PARAMETERS**: la richiesta non è valida per non conformità dei parametri in ingresso;
- **MISSING\_PLATE**: la richiesta non è valida: la targa non è stata precedentemente comunicata con *SendMeansOfTransportLoaded*.

### 1.1.8 SendMma

Web Service dedicato alla trasmissione dei manifesti di arrivo. Per l'implementazione del WS sono state adattate alcune sezioni di codice in modo utilizzarle a fattor comune con l'upload di un eMMA lato interfaccia:

- Step di acquisizione file
- Verifica se necessaria condivisione file verso il terminal
- Condivisione eventi verso Standard platform.

Il profilo utente che può utilizzare questo web service è: **CAD**.

Il messaggio da passare al web service ha questa struttura:

Struttura di VPCSPayload	Tipo	Descrizione
credentials		
data	version	string Versione del tracciato del file che si sta inviando
	fileName	string Nome del file di manifesto (compresa estensione)
	base64Content	string File testuale o XML del manifesto, codificato in stringa base64
	terminalCode	string Codice VPCS del terminal di riferimento
	agencyCode	string Codice VPCS dell'agenzia di riferimento

In caso di errore il web service può ritornare uno dei possibili codici:

- **CREDENTIAL\_ERROS**: credenziali fornite non riconosciute;
- **NOT\_ALLOWED**: l'utente non è autorizzato ad usare il servizio;
- **FILENAME\_ERROR**: nome file non assegnato / estensione non prevista;
- **TERMINAL\_ERROR**: codice terminal assente;
- **AGENCY\_ERROR**: codice agenzia assente;
- **MISSING\_BASE64**: il file xml non è in base64;
- **EMMA\_CONTENT\_ERROR**: errore nel contenuto del manifesto.

### 1.1.9 SendMmaResult

Web Service dedicato alla trasmissione del file di esito dei manifesti di arrivo. Per l'implementazione del WS sono state adattate alcune sezioni di codice in modo utilizzarle a fattor comune con l'upload di un esito eMMA lato interfaccia:

- Verifica se necessaria condivisione file verso il terminal
- Verifica generazione id digitali per merce non in container
- Verifica necessità di creazione automatica delle A44Ter
- Verifica flusso semplificato BAPLIE
- Condivisione eventi verso Standard platform.

Il profilo utente che può utilizzare questo web service è: **CAD**.

Il messaggio da passare al web service ha questa struttura:

Struttura di VPCSPayload	Tipo	Descrizione	
credentials			
data	version	string	Versione del tracciato del file che si sta inviando
	fileName	string	Nome del file IRISP Mma (compresa estensione)
	base64Content	string	File testuale o XML dell'irisp codificato in stringa base64

In caso di errore il web service può ritornare uno dei possibili codici:

- **CREDENTIAL\_ERROS**: credenziali fornite non riconosciute;
- **NOT\_ALLOWED**: l'utente non è autorizzato ad usare il servizio;
- **FILENAME\_ERROR**: nome file non assegnato / estensione non prevista;
- **MISSING\_BASE64**: il file xml non è in base64;
- **EMMA\_CONTENT\_ERROR**: errore nel contenuto del manifesto;

### 1.1.10 SendDu

Per trasmettere la distinta di uscita. All'acquisizione del file saranno applicate le stesse regole dell'upload manuale, in termini di regole sul nome file e la struttura del contenuto.

Il profilo utente che può utilizzare questo web service è: **CAD e doganalista**.

Il messaggio da passare al web service ha questa struttura:

Struttura di VPCSPayload	Tipo	Descrizione
credentials		
data	version	string Versione del tracciato della distinta file che si sta inviando
	fileName	string Nome del file della distinta di uscita (compresa estensione)
	base64Content	string File distinta di uscita codificato in stringa base64
	base64AdditionalContent	string File codificato in stringa base64 per i codici di telaio (non obbligatorio)

In caso di errore il web service può ritornare uno dei possibili codici:

- **CREDENTIAL\_ERRORS**: credenziali fornite non riconosciute;
- **NOT\_ALLOWED**: l'utente non è autorizzato ad usare il servizio;
- **FILENAME\_ERROR**: nome file non assegnato / estensione non prevista;
- **MISSING\_PARAMETERS**: file in base64 assente;

### 1.1.11 SendLandingFile

Per le tipologie di merce il cui sbarco viene già gestito tramite file COARRI in formato EDIFACT (contenitori, trailer, merce varia), è esposto un web service finalizzato a ricevere il file codificato in formato base64. Il nuovo web service si focalizza solo nel recepire il file e memorizzarlo nel server (sostituendosi quindi al FTP); verrà processato secondo la schedulazione dei file COARRI.

Il profilo utente che può utilizzare questo web service è: **terminalista**.

Il messaggio da passare al web service ha questa struttura:

Struttura di VPCSPayload	Tipo	Descrizione
credentials		
data	version	string(3) Versione del tracciato del file che si sta inviando [F]

fileName	string(150)	Nome del file (compresa estensione, edi o txt) [O]
base64Content	string	File EDI codificato in stringa base64 [O]

In caso di errore il web service può ritornare uno dei possibili codici:

- **CREDENTIAL\_ERROS**: credenziali fornite non riconosciute;
- **NOT\_ALLOWED**: l'utente non è autorizzato ad usare il servizio;
- **FILENAME\_ERROR**: problema con il nome file (non assegnato, caratteri non validi, estensione non prevista);
- **FILE\_NOT\_SAVED**: file non memorizzato;

### 1.1.12 SendTerminalGateOutFile

Per le tipologie di merce la cui uscita viene già gestita tramite file CODECO EDIFACT (contenitori, trailer, merce varia), è esposto un web service finalizzato a ricevere il file, codificato in formato base64. Il nuovo web si focalizzerà nel recepire il file e a memorizzarlo nel server (sostituendosi quindi al FTP).

Il profilo utente che può utilizzare questo web service è: **terminalista**.

Il messaggio da passare al web service ha questa struttura:

Struttura di VPCSPayload	Tipo	Descrizione
credentials		
data	version	string(3) Versione del tracciato del file che si sta inviando [F]
	fileName	Nome del file (compresa estensione, edi o txt) [O]
	base64Content	File EDI codificato in stringa base64 [O]

In caso di errore il web service può ritornare uno dei possibili codici di errore:

- **CREDENTIAL\_ERROS**: credenziali fornite non riconosciute;
- **NOT\_ALLOWED**: l'utente non è autorizzato ad usare il servizio;
- **FILENAME\_ERROR**: problema con il nome file (non assegnato, caratteri non validi, estensione non prevista);
- **FILE\_NOT\_SAVED**: file non memorizzato.