



ΥΠΟΥΡΓΕΙΟ ΨΗΦΙΑΚΗΣ ΔΙΑΚΥΒΕΡΝΗΣΗΣ

ΓΕΝΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

& ΨΗΦΙΑΚΗΣ ΔΙΑΚΥΒΕΡΝΗΣΗΣ

Κέντρο Διαλειτουργικότητας

**Κοινός Οδηγός Υλοποίησης -Τεχνικές Προδιαγραφές
για Φορείς Καταναλωτές Διαδικτυακών Υπηρεσιών
(client)**

**στο Κέντρο Διαλειτουργικότητας
του Υπουργείου Ψηφιακής Διακυβέρνησης**

Version 1.5 - 10/01/2025



Πίνακας Περιεχομένων

Περιεχόμενα

Πίνακας Περιεχομένων	2
1. Γενικά Στοιχεία	3
2. Αρχές Υλοποίησης των Web Services στο Κέντρο Διαλειτουργικότητας.....	5
2.1. Κοινές Υπηρεσίες.....	5
2.1.1 Αυθεντικοποίηση χρηστών	5
2.1.2 Έλεγχος Δικαιωμάτων Πρόσβασης Χρηστών	6
2.1.3 Ιχνηλασιμότητα (Auditing)	6
2.1.3.1 Στοιχεία - Εγγραφή Ιχνηλασιμότητας	6
2.1.3.2 Περιγραφή της Εγγραφής Ιχνηλασιμότητας (auditRecord).....	6
2.1.4 Επαλήθευση Ψηφιακής Υπογραφής.....	8
2.1.5 Διαχείριση Εξαιρέσεων	8
3. Ασφάλεια	9
3.1 SOAP SERVICES.....	9
3.3 REST SERVICES.....	10
4. Πεδία εισόδου διαδικτυακών υπηρεσιών	11
4.1 SOAP SERVICES.....	11
4.2 REST SERVICES.....	11
5. Πεδία εξόδου διαδικτυακών υπηρεσιών.....	12
5.1 Πεδία εξόδου με τιμές NULL.....	14
6. Αντιμετώπιση Λαθών	15
7. Γενική δομή αιτήματος κλήσης των διαδικτυακών υπηρεσιών	16
7.1 SOAP SERVICES.....	16
7.2 REST SERVICES.....	17
8. Γενική δομή απόκρισης των διαδικτυακών υπηρεσιών	18
8.1 SOAP SERVICES.....	18
8.2 REST SERVICES.....	19
9. Προαπαιτούμενα για κατανάλωση διαδικτυακών υπηρεσιών του ΚΕΔ	20
ΠΑΡΑΡΤΗΜΑ Α : Αναλυτική περιγραφή πεδίων Εγγραφής Ιχνηλασιμότητας	21
ΠΑΡΑΡΤΗΜΑ Β : Παραδείγματα κλήσης & απάντησης διαδικτυακών υπηρεσιών	22



1. Γενικά Στοιχεία

Οι διαδικτυακές υπηρεσίες (web services) είναι μία τεχνολογία που επιτρέπει στα πληροφοριακά συστήματα - εφαρμογές να επικοινωνούν μεταξύ τους ανεξαρτήτως γλώσσας προγραμματισμού και πλατφόρμας. Μία διαδικτυακή υπηρεσία είναι μία διεπαφή λογισμικού (software interface) που περιλαμβάνει: α) μία συλλογή από λειτουργίες, οι οποίες περιγράφουν τον τρόπο ανταλλαγής μηνυμάτων, που βασίζονται στο μορφότυπο XML (Extensible Markup Language) και β) τα δεδομένα προς ανταλλαγή με άλλη εφαρμογή.

Οι διαδικτυακές υπηρεσίες παρέχουν λειτουργικότητα σε χρήστες του διαδικτύου μέσα από ένα δικτυακό πρωτόκολλο το οποίο σε αρκετές περιπτώσεις είναι το SOAP (Simple Object Access Protocol). Οι SOAP διαδικτυακές υπηρεσίες προσφέρουν έναν λεπτομερή τρόπο περιγραφής των διεπαφών τους και επιτρέπουν στον τελικό χρήστη-καταναλωτή (consumer) να αναπτύξει μια εφαρμογή-πελάτη (client application) για να επικοινωνήσει μαζί τους. Η τεχνική περιγραφή ενός SOAP web service παρέχεται σε ένα έγγραφο XML το οποίο ονομάζεται έγγραφο WSDL (Web Services Description Language).

Το έγγραφο WSDL μιας SOAP διαδικτυακής υπηρεσίας στο Κέντρο Διαλειτουργικότητας του Υπουργείου Οικονομικών παρέχεται μέσω ενός URL, όπως παρακάτω:

Πιλοτικό περιβάλλον: <https://test.gsis.gr/esbpilot/testService?wsdl>

Παραγωγικό περιβάλλον: <https://ked.gsis.gr/esb/testService?wsdl>

Τα τελευταία χρόνια έχουν γίνει πολύ δημοφιλείς οι διαδικτυακές υπηρεσίες RESTful οι οποίες βασίζονται στον ιστό και επιτρέπουν την αλληλεπίδραση μεταξύ πελατών και διακομιστών μέσω του πρωτοκόλλου HTTP. Το REST (Representational State Transfer) είναι ένα αρχιτεκτονικό στυλ για τη σχεδίαση διαδικτυακών εφαρμογών που έχει τα εξής κύρια χαρακτηριστικά:

- **Client-Server:** Ο διακομιστής παρέχει πόρους και ο πελάτης τους καταναλώνει. Αυτή η διαφοροποίηση βοηθά στη βελτίωση της επεκτασιμότητας και της φορητότητας
- **Stateless:** Κάθε αίτημα από τον πελάτη προς τον διακομιστή πρέπει να περιλαμβάνει όλες τις απαραίτητες πληροφορίες για την εξυπηρέτηση του αιτήματος. Ο διακομιστής δεν αποθηκεύει καμία πληροφορία για τον πελάτη μεταξύ των αιτημάτων



- **Cacheable:** Οι απαντήσεις από τον διακομιστή μπορούν να αποθηκεύονται στην κρυφή μνήμη για τη βελτίωση της απόδοσης και της αποδοτικότητας
- **Uniform Interface:** Το REST προωθεί τη χρήση μιας ομοιόμορφης διεπαφής που επιτρέπει την απλοποίηση και τη χαλάρωση των αλληλεπιδράσεων μεταξύ πελάτη και διακομιστή
- **Resources:** Οι πόροι αντιπροσωπεύονται συνήθως από URLs (Uniform Resource Locators). Κάθε URL αντιστοιχεί σε έναν συγκεκριμένο πόρο, όπως μια οντότητα
- **HTTP methods:** Το REST χρησιμοποιεί τις HTTP μεθόδους (GET, POST, PUT, DELETE κλπ) για το χειρισμό των πόρων
- **Status Codes:** Το REST χρησιμοποιεί τους κωδικούς κατάστασης HTTP για να ενημερώνει τον πελάτη για το αποτέλεσμα ενός αιτήματος. Για παράδειγμα: STATUS 200/OK, STATUS 403/Forbidden, STATUS 404/Not Found, STATUS 500/Internal Server Error
- **Flexible Representation Formats:** Τα δεδομένα μπορούν να ανταλλαχθούν σε διάφορες μορφές, όπως XML (Extensible Markup Language), HTML (Hypertext Markup Language), CSV (Comma-Separated Values), JSON (JavaScript Object Notation) ή Plain Text (Απλό Κείμενο). Το JSON είναι το πιο δημοφιλές format λόγω της απλότητας και της καλύτερης απόδοσης και είναι αυτό που χρησιμοποιείται στα REST web services του Κέντρου Διαλειτουργικότητας

Τα πλεονεκτήματα της αρχιτεκτονικής REST είναι τα παρακάτω:

- **Απλότητα:** Η χρήση των κανονικών μεθόδων HTTP το καθιστά εύκολα κατανοητό και υλοποιήσιμο και πολύ απλό στη χρήση
- **Ανεξαρτησία:** Οι πελάτες και οι διακομιστές μπορούν να αναπτύσσονται ανεξάρτητα ο ένας από τον άλλο
- **Επεκτασιμότητα:** Τα REST APIs είναι εξαιρετικά κλιμακούμενα καθώς η REST αρχιτεκτονική υποστηρίζει την επεκτασιμότητα μέσω της διάσπασης της αρχιτεκτονικής σε πελάτη και διακομιστή
- **Ευελιξία:** Τα REST APIs μπορούν να επικοινωνούν χρησιμοποιώντας οποιαδήποτε μορφή δεδομένων (όχι μόνο XML που χρησιμοποιείται στα SOAP services)
- **Αξιοπιστία & Ασφάλεια:** Η αρχιτεκτονική του REST μειώνει την πολυπλοκότητα και αυξάνει την ευελιξία και την ασφάλεια των συστημάτων

Η τεχνική περιγραφή μιας REST υπηρεσίας του ΚΕΔ μπορεί να αντληθεί κάνοντας μια GET κλήση στο ENDPOINT της υπηρεσίας, για παράδειγμα:

Πιλοτικό περιβάλλον: <https://test.gsis.gr/esbpilot/testRestService/getData>



Παραγωγικό περιβάλλον: <https://ked.gsis.gr/esb/testRestService/getData>
Και οι διαθέσιμες λειτουργίες (APIs) μιας rest υπηρεσίας μπορούν να αντληθούν από το:
Πιλοτικό περιβάλλον: https://test.gsis.gr/esbpilot/testRestService?_wadl
Παραγωγικό περιβάλλον: https://ked.gsis.gr/esb/testRestService?_wadl

2. Αρχές Υλοποίησης των Web Services στο Κέντρο Διαλειτουργικότητας

2.1. Κοινές Υπηρεσίες

Στο Κέντρο Διαλειτουργικότητας έχει υλοποιηθεί ένα σύνολο κοινών υπηρεσιών. Οι κοινές υπηρεσίες περιλαμβάνουν τις παρακάτω λειτουργίες:

- ✓ Αυθεντικοποίηση Χρηστών (User Authentication)
- ✓ Έλεγχο Δικαιωμάτων Πρόσβασης Χρηστών (User Authorization)
- ✓ Υπηρεσία Ιχνηλασιμότητας (Auditing)
- ✓ Επαλήθευση Ψηφιακής Υπογραφής (Digital Signature Verification)
- ✓ Διαχείριση Εξαίρέσεων (Exception Handling)

Οι κοινές υπηρεσίες (Common Services) έχουν υλοποιηθεί στο Enterprise Service Bus (ESB) και χρησιμοποιούνται από όλες τις διαδικτυακές υπηρεσίες του Κέντρου Διαλειτουργικότητας. Η υλοποίηση των παραπάνω υπηρεσιών συμβάλλει στην υιοθέτηση αρθρωτής αρχιτεκτονικής και προτυποποιεί την ανάπτυξη των διαδικτυακών υπηρεσιών, διαθέτοντας τις παραπάνω λειτουργίες για χρήση από όλες τις διαδικτυακές υπηρεσίες.

2.1.1 Αυθεντικοποίηση χρηστών

Η υπηρεσία αυτή περιλαμβάνει πιστοποίηση του ονόματος χρήστη (username) και κωδικού πρόσβασης (password) που χρησιμοποιεί η διαδικτυακή υπηρεσία. Τα παραπάνω διαπιστευτήρια εκδίδονται μέσω της Εφαρμογής Διαχείρισης Αιτημάτων Διαλειτουργικότητας (ΕΔΑ).



2.1.2 Έλεγχος Δικαιωμάτων Πρόσβασης Χρηστών

Η υπηρεσία αυτή ελέγχει αν ο (αυθεντικοποιημένος) χρήστης έχει δικαίωμα χρήσης της συγκεκριμένης διαδικτυακής υπηρεσίας που καλεί.

2.1.3 Ιχνηλασιμότητα (Auditing)

Η υπηρεσία ιχνηλασιμότητας αποτελεί έναν μηχανισμό καταγραφής πληροφοριών για όλες τις διαδικτυακές υπηρεσίες του Κέντρου Διαλειτουργικότητας. Λαμβάνει χώρα σε κάθε κλήση διαδικτυακής υπηρεσίας, μετά από την επιτυχημένη αυθεντικοποίηση χρήστη. Η υπηρεσία ιχνηλασιμότητας περιλαμβάνει:

- Έλεγχο για την ύπαρξη των απαραίτητων πεδίων audit σε κάθε κλήση.
- Δημιουργία ενός μοναδικού αριθμού εξυπηρέτησης κλήσης καθώς και της ημερομηνίας κλήσης (callSequenceId και callSequenceDate).
- Λήψη των δεδομένα εισόδου (Request XML) του αιτήματος κλήσης και καταγραφή των απαραίτητων στοιχείων της.

2.1.3.1 Στοιχεία - Εγγραφή Ιχνηλασιμότητας

Όλες οι διαδικτυακές υπηρεσίες απαιτούν κατά την κλήση τους την προσθήκη των **στοιχείων (εγγραφής) ιχνηλασιμότητας (auditRecord)**. Τα στοιχεία αυτά χρησιμοποιούνται προκειμένου να γίνεται καταγραφή της κάθε κλήσης (request call) αλλά και για να είναι εφικτή η ιχνηλασιμότητα των κλήσεων που έχουν εξυπηρετηθεί. Τα στοιχεία ιχνηλασιμότητας (auditRecord) περιέχουν τα παρακάτω 6 πεδία:

1. **auditTransactionId**: αλφαριθμητικό – α/α κλήσης της εφαρμογής πελάτη του Φορέα (π.χ. “1”, “2” κλπ)
2. **auditTransactionDate**: ημερομηνία & ώρα, της μορφής yyyy-mm-ddThh:mm:ssZ – η ημερομηνία & ώρα κλήσης (π.χ. “2016-09-21T10:08:24Z”)
3. **auditUnit**: αλφαριθμητικό – όνομα φορέα που καλεί τη διαδικτυακή υπηρεσία (π.χ. “ΙΚΑ” ή “Δήμος Αθηναίων”)
4. **auditProtocol**: αλφαριθμητικό – αριθμός πρωτοκόλλου (π.χ. “1123/10-10-2016”)
5. **auditUserId**: αλφαριθμητικό – αναγνωριστικό τελικού χρήστη που καλεί τη διαδικτυακή υπηρεσία (π.χ. “user1”)
6. **auditUserIp**: αλφαριθμητικό – η διεύθυνση IP του τελικού χρήστη που καλεί τη διαδικτυακή υπηρεσία (π.χ. “10.18.24.13”)

2.1.3.2 Περιγραφή της Εγγραφής Ιχνηλασιμότητας (auditRecord)

Η δομή της εγγραφής ιχνηλασιμότητας ορίζεται ως εξής:



- WSDL περιγραφή (SOAP services)

```
<xs:complexType name="auditRecord">
  <xs:sequence>
    <xs:element name="auditTransactionId" type="xs:string"/>
    <xs:element name="auditTransactionDate" type="xs:dateTime"/>
    <xs:element minOccurs="0" name="auditUnit" type="xs:string"/>
    <xs:element minOccurs="0" name="auditProtocol" type="xs:string"/>
    <xs:element minOccurs="0" name="auditUserId" type="xs:string"/>
    <xs:element minOccurs="0" name="auditUserIp" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

- Περιγραφή με χρήση JSON Schema (REST services)

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "properties": {
    "auditRecord": {
      "type": "object",
      "properties": {
        "auditTransactionId": {
          "type": "string",
          "description": "Unique request code (unique identifier for the current request/call)"
        },
        "auditTransactionDate": {
          "type": "string",
          "format": "date-time",
          "description": "Date and time of the transaction/request (ISO 8601 format)"
        },
        "auditUnit": {
          "type": "string",
          "description": "Name/description of the entity or sub-entity"
        },
        "auditProtocol": {
          "type": "string",
          "description": "Unique protocol number or description"
        },
        "auditUserId": {
          "type": "string",
          "description": "Name of the end user (username or user ID) "
        }
      }
    }
  }
}
```



```
"auditUserIp": {  
  "type": "string",  
  "description": "The IP address (IPv4 or IPv6) of the end user"  
}  
,  
"required": ["auditTransactionId", "auditTransactionDate"]  
}  
,  
"required": ["auditRecord"]  
}
```

Τα δύο πρώτα πεδία (auditTransactionId και auditTransactionDate) είναι υποχρεωτικά, ενώ από τα υπόλοιπα τέσσερα πεδία είναι υποχρεωτικό τουλάχιστον ένα (ορίζονται από το Διαχειριστή της Διαδικτυακής υπηρεσίας μέσα από την Εφαρμογή Διαχείρισης Αιτημάτων Διαλειτουργικότητας (ΕΔΑ)). Συνίσταται κατά την κλήση των λειτουργιών των διαδικτυακών υπηρεσιών να συμπληρώνονται **όλα** τα παραπάνω πεδία.

Επιπλέον αποτελεί **υποχρέωση του κάθε Φορέα** να τηρεί το ημερολόγιο καταγραφής κλήσεων που πρέπει να περιλαμβάνει όλα τα παραπάνω στοιχεία για κάθε κλήση.

Περισσότερες πληροφορίες παρέχονται στο Παράρτημα Α.

2.1.4 Επαλήθευση Ψηφιακής Υπογραφής

Η υπηρεσία επαλήθευσης ψηφιακής υπογραφής μηνύματος διενεργεί την επαλήθευση της ορθότητας του υπογεγραμμένου μηνύματος που λαμβάνεται από τη διαδικτυακή υπηρεσία. Το κύκλωμα της ψηφιακής υπογραφής/επαλήθευσης είναι προαιρετικό και ενεργοποιείται κατά περίπτωση.

2.1.5 Διαχείριση Εξαιρέσεων

Η υπηρεσία διαχείρισης εξαιρέσεων έχει υλοποιηθεί μέσω ενός μηχανισμού χειρισμού και καταγραφής αναπάντεχων συμβάντων (εξαιρέσεων) ο οποίος έχει τη δυνατότητα να εντοπίζει και να καταγράφει οποιοδήποτε «έκτακτο» πρόβλημα (exception) δημιουργηθεί κατά τη διάρκεια της εξυπηρέτησης της κλήσης μιας διαδικτυακής υπηρεσίας στο Κέντρο Διαλειτουργικότητας.



3. Ασφάλεια

3.1 SOAP SERVICES

Οι SOAP διαδικτυακές υπηρεσίες του Κέντρου Διαλειτουργικότητας χρησιμοποιούν WS-Security πρωτόκολλο¹ και η εξουσιοδότηση στις υπηρεσίες τους γίνεται με χρήση διαπιστευτηρίων-συνθηματικών (username και password). Τα συνθηματικά δημιουργούνται στην Εφαρμογή Διαχείρισης Αιτημάτων Διαλειτουργικότητας (ΕΔΑ).

Κάθε κλήση (call request) μιας λειτουργίας διαδικτυακών υπηρεσιών που πραγματοποιείται στο Κέντρο Διαλειτουργικότητας θα πρέπει να περιλαμβάνει τον SOAP Header στον οποίο δηλώνονται τα παραπάνω διαπιστευτήρια (username και password), και επιπρόσθετα στοιχεία που αφορούν στη συγκεκριμένη κλήση.

Ένα παράδειγμα SOAP Header μιας κλήσης (call request) είναι το παρακάτω:

```
<soapenv:Header>
  <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
    <wsse:UsernameToken wsu:Id="UsernameToken-134F28397E84840D3F14484484569051">
      <wsse:Username>myUsername</wsse:Username>
      <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-
token-profile-1.0#PasswordText">myPassword</wsse:Password>
    </wsse:UsernameToken>
  </wsse:Security>
</soapenv:Header>
```

Σε περίπτωση που παραλειφθεί το SOAP Header ή που δεν δοθούν σωστά συνθηματικά τότε η SOAP διαδικτυακή υπηρεσία απαντάει με http status 500 (Internal Server Error) και επιστρέφει κατάλληλο μήνυμα λάθους:

“A security error was encountered when verifying the message” (με κωδικό **“SecurityError”**)

¹ Το πρωτόκολλο **WS-security** καθορίζει τον τρόπο με τον οποίο μπορεί να επιβληθεί η ακεραιότητα και η εμπιστευτικότητα σε μηνύματα και επιτρέπει την επικοινωνία διαφόρων μορφών token ασφαλείας, όπως το Security Assertion Markup Language (SAML), Kerberos, και X.509. Κύριος στόχος του είναι η χρήση της XML υπογραφής και κρυπτογράφησης (XML Signature και XML Encryption) για την παροχή ασφάλειας από σημείο σε σημείο (end-to-end security).



3.3 REST SERVICES

Ως μέθοδος αυθεντικοποίησης στα REST services του ΚΕΔ μπορεί να χρησιμοποιηθεί είτε Basic είτε Bearer Authorization. Στο παρόν εγχειρίδιο περιγράφεται ο τρόπος αυθεντικοποίησης μέσω Basic Authorization.

Αυθεντικοποίηση μέσω Basic Authorization

Κάθε POST κλήση που πραγματοποιείται από τον client σε ένα REST service, θα πρέπει να περιλαμβάνει έναν HTTP Header ο οποίος θα περιέχει το όνομα του χρήστη και τον κωδικό πρόσβασης σε κωδικοποιημένη μορφή. Ο Header πρέπει να έχει την εξής μορφή:

Authorization: Basic <διαπιστευτήρια>

όπου τα διαπιστευτήρια είναι η κωδικοποίηση σε Base64 του ονόματος χρήστη και του κωδικού πρόσβασης, συνενωμένα με το χαρακτήρα ":". Για παράδειγμα, αν το όνομα χρήστη είναι "user" και ο κωδικός πρόσβασης είναι "password1", τότε θα πρέπει να συμπεριληφθεί ο παρακάτω Header ("Authorization") για την αυθεντικοποίηση:

Authorization: Basic dXNlcjpwYXNzd29yZDE=

όπου "dXNlcjpwYXNzd29yZDE=" είναι η Base64 κωδικοποίηση του αλφαριθμητικού "user:password1".

Σε περίπτωση που δοθούν λανθασμένα συνθηματικά (δηλαδή λάθος username ή password), τότε επιστρέφεται σχετικό μήνυμα λάθους. Συγκεκριμένα, στη δομή errorRecord επιστρέφεται το μήνυμα «**Η αυθεντικοποίηση χρήστη απέτυχε (λανθασμένα συνθηματικά)**» με κωδικό «**GEN_AUTHENTICATION_FAILED**»

Να επισημανθεί πως όταν η αυθεντικοποίηση αποτύχει, το HTTP STATUS που επιστρέφεται από την κλήση του rest service είναι 200 και όχι 401, όπως συνηθίζεται στις περιπτώσεις αποτυχημένου authentication. Γενικότερα όλες οι κλήσεις προς τα rest services του Κέντρου Διαλειτουργικότητας επιστρέφουν HTTP STATUS 200 και η διαχείριση λαθών γίνεται με βάση τον κωδικό και την περιγραφή σφάλματος της δομής errorRecord.



4. Πεδία εισόδου διαδικτυακών υπηρεσιών

4.1 SOAP SERVICES

Κάθε λειτουργία των SOAP διαδικτυακών υπηρεσιών (SOAP web service operation) που περιγράφεται στο WSDL περιλαμβάνει τα παρακάτω πεδία εισόδου στο σώμα κλήσης:

- SoapHeader: είναι η SOAP κεφαλίδα του μηνύματος στην οποία παρέχονται τα διαπιστευτήρια κλήσης της διαδικτυακής υπηρεσίας,
- auditRecord: η εγγραφή ιχνηλασιμότητας,
- requestRecord: αποτελεί το ουσιαστικό κομμάτι της κλήσης διαδικτυακής υπηρεσίας. Περιλαμβάνει τα κατά περίπτωση πεδία εισόδου επιχειρησιακών δεδομένων της λειτουργίας

4.2 REST SERVICES

Αντιστοίχως, κάθε λειτουργία των REST διαδικτυακών υπηρεσιών (REST API) περιλαμβάνει τις παρακάτω δομές/εγγραφές μέσα στο σώμα κλήσης (json request):

- auditRecord: η εγγραφή ιχνηλασιμότητας,
- requestRecord: αποτελεί το ουσιαστικό κομμάτι της κλήσης διαδικτυακής υπηρεσίας. Περιλαμβάνει τα κατά περίπτωση πεδία εισόδου επιχειρησιακών δεδομένων της λειτουργίας

Στις κλήσεις των REST APIs, τα διαπιστευτήρια ΔΕΝ περιλαμβάνονται μέσα στο σώμα κλήσης (δηλαδή μέσα στο body/JSON Request του αιτήματος) αλλά συμπληρώνονται στον "Authorization" header της POST κλήσης, όπως προδιαγράφηκε προηγουμένως.



5. Πεδία εξόδου διαδικτυακών υπηρεσιών

Κάθε λειτουργία των διαδικτυακών υπηρεσιών (SOAP ή REST) επιστρέφει τα παρακάτω πεδία εξόδου:

- **callSequenceId**: αποτελεί το μοναδικό αριθμό αναγνώρισης της εξυπηρέτησης κλήσης της διαδικτυακής υπηρεσίας (web service request). Ο αριθμός αυτός παράγεται αυτόματα κατά την έναρξη της κλήσης και είναι μοναδικός για όλες τις κλήσεις στο Κέντρο Διαλειτουργικότητας,
- **callSequenceDate**: καταγράφει την ημερομηνία και την ώρα κλήσης. Ουσιαστικά αποτελεί τη χρονοσήμανση (timestamp) του εξυπηρετητή του Κέντρου Διαλειτουργικότητας κατά την οποία αποδόθηκε το callsequenceid κλήσης και αποτελεί ζευγάρι με αυτό,
- **errorRecord**: αποτελεί μια δομή στην οποία καταγράφονται τα σφάλματα που προκύπτουν κατά την εξυπηρέτηση της κλήσης και περιέχει τα πεδία **errorCode** (κωδικός σφάλματος) και **errorDescr** (περιγραφή μηνύματος σφάλματος). Σε περίπτωση που δεν συμβεί κάποιο σφάλμα (τεχνικό/λογικό) τότε τα πεδία της δομής επιστρέφουν με τιμή NULL

Η δομή των παραπάνω πεδίων είναι η εξής:

- WSDL περιγραφή (SOAP services)

```
<xs:element name="callSequenceId" type="xs:decimal"/>
<xs:element name="callSequenceDate" type="xs:dateTime"/>
<xs:element name="errorRecord" type="tns:errorRecord"/>

<xs:complexType name="errorRecord">
  <xs:sequence>
    <xs:element minOccurs="0" name="errorCode" type="xs:string"/>
    <xs:element minOccurs="0" name="errorDescr" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```



- Περιγραφή με χρήση JSON Schema (REST services)

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "properties": {
    "callSequenceId": {
      "type": "number"
    },
    "callSequenceDate": {
      "type": "string",
      "format": "date-time"
    },
    "errorRecord": {
      "type": "object",
      "properties": {
        "errorCode": {
          "type": "string"
        },
        "errorDescr": {
          "type": "string"
        }
      }
    }
  }
},
"required": [
  "callSequenceId", "callSequenceDate", "errorRecord"
]
}
```

Εκτός από τα παραπάνω πεδία, κάθε λειτουργία διαδικτυακής υπηρεσίας (SOAP service operation/REST API) επιστρέφει και τα πεδία των εκάστοτε επιχειρησιακών δεδομένων της λειτουργίας (τα πεδία εξαρτώνται από τη συγκεκριμένη λειτουργία).



5.1 Πεδία εξόδου με τιμές NULL

Γενικά υπάρχουν δύο τρόποι για την αναπαράσταση των πεδίων που έχουν τιμή NULL στην απάντηση (XML Response για τα SOAP Services ή JSON Response για τα REST Services) της υπηρεσίας.

Ο πρώτος τρόπος είναι να μην εμφανίζονται καθόλου τα πεδία με τιμή NULL στο Response ενώ ο δεύτερος τρόπος είναι να εμφανίζονται/αποτυπώνονται ρητά τα πεδία στην έξοδο, έχοντας στο αντίστοιχο XML tag τους την ιδιότητα (attribute) `nil="true"` για τα SOAP Services και έχοντας κενό περιεχόμενο (`null`) για τα REST Services.

Παράδειγμα:

- SOAP service

```
<fatherName xsi:nil="true" />
```

- REST service

```
{  
  "fatherName": null  
}
```

Στις διαδικτυακές υπηρεσίες του Κέντρου Διαλειτουργικότητας έχει υιοθετηθεί ο πρώτος τρόπος. Επομένως, η απουσία των πεδίων εξόδου από το Response σημαίνει αυτομάτως ότι αυτά έχουν τιμή NULL.



6. Αντιμετώπιση Λαθών

Σε περίπτωση λάθους κατά την κλήση μιας διαδικτυακής υπηρεσίας του Κέντρου Διαλειτουργικότητας, στην απάντηση (XML Response ή JSON Response) επιστρέφεται ο κωδικός και το μήνυμα λάθους που αποτυπώνονται από τα πεδία **errorCode** και **errorDescr**.

Στον παρακάτω πίνακα καταγράφονται οι πιο συνηθισμένες περιπτώσεις λαθών.

Κωδικός λάθος (Error Code Name)	Περιγραφή μηνύματος λάθους
GEN_INVALID_STRUCTURE_FORMAT*	Η δομή δεν είναι σύμφωνη με τις προδιαγραφές της υπηρεσίας. Τουτέστιν, τα πεδία εισόδου δεν μεταβιβάστηκαν στην κατάλληλη μορφή (σφάλμα στο JSON request ή μεταβίβαση μη αναμενόμενων πεδίων)
GEN_AUTHENTICATION_FAILED*	Η αυθεντικοποίηση χρήστη απέτυχε, δηλαδή δόθηκαν λανθασμένα συνθηματικά (λάθος username ή/και password) ή χρησιμοποιήθηκαν τα συνθηματικά του πιλοτικού περιβάλλοντος στο παραγωγικό περιβάλλον (ή αντιστρόφως)
GEN_AUDIT_VALIDATION_INV_DATA_OR_UNAUTH_OPER	Τα διαπιστευτήρια (username & password) που χρησιμοποιήθηκαν για την αυθεντικοποίηση δεν έχουν εξουσιοδότηση για χρήση της συγκεκριμένης διαδικτυακής υπηρεσίας
GEN_AUDIT_VALIDATION_INVALID_AUDIT_DATA	Παρουσιάστηκε σφάλμα κατά τον έλεγχο των audit πεδίων της κλήσης. Κάποια από τα απαιτούμενα πεδία της εγγραφής ιχνηλασιμότητας λείπουν ή είναι κενά (τα συγκεκριμένα πεδία περιγράφονται στο πεδίο errorDescr)
GEN_INVALID_DATA	Παρουσιάστηκε σφάλμα κατά τον έλεγχο εγκυρότητας των πεδίων εισόδου της υπηρεσίας. Κάποιο από τα υποχρεωτικά πεδία δεν δόθηκε ή δόθηκε με μη έγκυρη τιμή (στο πεδίο errorDescr αναφέρονται πιο συγκεκριμένες λεπτομέρειες για το σφάλμα)
GEN_COMMUNICATION_ERROR	Παρουσιάστηκε σφάλμα κατά την επικοινωνία με τη διαδικτυακή υπηρεσία του εξωτερικού φορέα (επικοινωνήστε με την Τεχνική Υποστήριξη του εξωτερικού φορέα)
GEN_DATABASE_CONNECTION_ERROR	Παρουσιάστηκε σφάλμα κατά τη σύνδεση με τη Βάση Δεδομένων (επικοινωνήστε με την Τεχνική Υποστήριξη του ΚΕΔ)
GEN_GENERAL_ERROR	Παρουσιάστηκε απροσδιόριστο σφάλμα (επικοινωνήστε με την Τεχνική Υποστήριξη του ΚΕΔ)
GEN_AUTHORIZATION_MAX_OPEN_CALLS_EXCEEDED	Έχει γίνει υπέρβαση του ημερήσιου ορίου κλήσεων (Authorization Failed)

* οι κωδικοί GEN_INVALID_STRUCTURE_FORMAT & GEN_AUTHENTICATION_FAILED αφορούν μόνο τα REST Services



7. Γενική δομή αιτήματος κλήσης των διαδικτυακών υπηρεσιών

7.1 SOAP SERVICES

Κάθε μέθοδος/λειτουργία των SOAP υπηρεσιών του Κέντρου Διαλειτουργικότητας απαιτεί ως είσοδο (input) τα παρακάτω πεδία στο XML Request της κλήσης:

* **εγγραφή ιχνηλασιμότητας** (βλέπε παράγραφο 2.1.3)

* **εγγραφή πεδίων εισόδου της λειτουργίας**. Το όνομα της εγγραφής περιλαμβάνει ένα κομμάτι της ονομασίας της λειτουργίας και έχει ως κατάληξη τη λέξη "InputRecord" (για παράδειγμα "oaedHistoryInputRecord", "getIdentityInputRecord" κλπ). Περιλαμβάνει επίσης και τα κατά περίπτωση πεδία των επιχειρησιακών δεδομένων.

Οι παραπάνω δύο εγγραφές "εσωκλείονται" σε μια γενική δομή, το όνομα της οποίας ορίζεται από το όνομα της λειτουργίας και έχει ως κατάληξη τη λέξη "Request" (για παράδειγμα "oaedHistoryRequest", "getIdentityRequest" κλπ).

Ακολουθεί ένα παράδειγμα κλήσης (XML Request) για τη λειτουργία *getVerificationProgress* της διαδικτυακής υπηρεσίας *verificationProgressService*:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ind="http://gsis.ggps.interoperability/verificationProgressService">
  <soapenv:Header>
    ...
  </soapenv:Header>
  <soapenv:Body>
    <getVerificationProgressRequest>
      <auditRecord> <!-- η εγγραφή του auditRecord -->
        <auditTransactionId> ... </auditTransactionId>
        <auditTransactionDate> ... </auditTransactionDate>
        <auditUnit> ... </auditUnit>
        <auditProtocol> ... </auditProtocol>
        <auditUserId> ... </auditUserId>
        <auditUserIp> ... </auditUserIp>
      </auditRecord>
      <getVerificationProgressInputRecord> <!-- η εγγραφή με τα πεδία εισόδου της υπηρεσίας -->
        <dateFrom> ... </dateFrom>
        <dateTo> ... </dateTo>
        <extOrg> ... </extOrg>
      </getVerificationProgressInputRecord>
    </getVerificationProgressRequest>
  </soapenv:Body>
</soapenv:Envelope>
```




7.2 REST SERVICES

Κάθε API των REST διαδικτυακών υπηρεσιών του Κέντρου Διαλειτουργικότητας απαιτεί ως είσοδο (input) τα παρακάτω πεδία στο σώμα (JSON Request) της κλήσης:

* **εγγραφή ιχνηλασιμότητας** (βλέπε παράγραφο 2.1.3)

* **εγγραφή πεδίων εισόδου της λειτουργίας**. Το όνομα της εγγραφής περιλαμβάνει ένα κομμάτι της ονομασίας της λειτουργίας και έχει ως κατάληξη τη λέξη "InputRecord" (για παράδειγμα "getStatusPDInputRecord", "getImageInputRecord" κλπ). Περιλαμβάνει επίσης και τα κατά περίπτωση πεδία των επιχειρησιακών δεδομένων.

Οι παραπάνω δύο εγγραφές "εσωκλείονται" μαζί σε ένα JSON Object.

Ακολουθεί ένα παράδειγμα κλήσης (JSON Request) για το API *getChildrenByParentAfm* της διαδικτυακής υπηρεσίας *familyInfoRestService*:

```
{
  "auditRecord": {
    "auditTransactionId": "1",
    "auditTransactionDate": "2024-08-26T15:18:30Z",
    "auditUnit": "GSIS",
    "auditProtocol": "1",
    "auditUserId": "user",
    "auditUserIp": "0.0.0.0"
  },
  "getChildrenByParentAfmInputRecord": {
    "afm": "000000000"
  }
}
```



8. Γενική δομή απόκρισης των διαδικτυακών υπηρεσιών

8.1 SOAP SERVICES

Τα πεδία που λαμβάνονται ως απόκριση της κλήσης (XML Response) κάθε λειτουργίας/μεθόδου των SOAP διαδικτυακών υπηρεσιών είναι τα ακόλουθα:

* **εγγραφή πεδίων εξόδου της λειτουργίας.** Το όνομα της εγγραφής περιλαμβάνει ένα κομμάτι της ονομασίας της λειτουργίας και έχει ως κατάληξη τη λέξη "OutputRecord" (για παράδειγμα "oaedHistoryOutputRecord", "getIdentityOutputRecord" κλπ). Η εγγραφή αυτή περιλαμβάνει τα κατά περίπτωση πεδία των επιχειρησιακών στοιχείων που επιστρέφονται.

* **πεδίο callSequenceId, πεδίο callSequenceDate, εγγραφή σφάλματος** (βλέπε παράγραφο 5).

Όλα τα παραπάνω πεδία "εσωκλείονται" σε μια γενική δομή, το όνομα της οποίας ορίζεται από την ονομασία της λειτουργίας και έχει ως κατάληξη τη λέξη "Response" (για παράδειγμα "oaedHistoryResponse", "getIdentityResponse" κλπ).

Ακολουθεί ένα παράδειγμα απόκρισης (XML Response) για τη λειτουργία *retrievePayProgress* της διαδικτυακής υπηρεσίας *payProgressService*:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:retrievePayProgressResponse xmlns:ns2=" http://gsis.ggps.interoperability/payProgressService">
      <retrievePayProgressOutputRecord> <!-- η εγγραφή με τα πεδία εξόδου της υπηρεσίας -->
        <deletedAmount> ... </deletedAmount>
        <firstEstablishedAmount> ... </firstEstablishedAmount>
        <overdueAmount> ... </overdueAmount>
        <paidAmount> ... </paidAmount>
        <reducedAmount> ... </reducedAmount>
        <restAmount> ... </restAmount>
        <totalAmount> ... </totalAmount>
      </retrievePayProgressOutputRecord>
      <callSequenceId>1234567890</callSequenceId>
      <callSequenceDate>2024-08-27T16:08:30.464+03:00</callSequenceDate>
      <errorRecord/> <!-- η εγγραφή του errorRecord, εδώ τα πεδία του errorDescr & errorCode
είναι null επειδή δεν συνέβη κάποιο σφάλμα -->
    </ns2:retrievePayProgressResponse>
  </soap:Body>
</soap:Envelope>
```



8.2 REST SERVICES

Τα πεδία που επιστρέφονται ως απόκριση της κλήσης (JSON Response) κάθε API των REST διαδικτυακών υπηρεσιών είναι τα ακόλουθα:

* **εγγραφή πεδίων εξόδου της λειτουργίας.** Το όνομα της εγγραφής περιλαμβάνει ένα κομμάτι της ονομασίας του REST API και έχει ως κατάληξη τη λέξη "OutputRecord" (για παράδειγμα "getStatusPDOutputRecord", "getImageOutputRecord" κλπ). Η εγγραφή αυτή περιλαμβάνει τα κατά περίπτωση πεδία των επιχειρησιακών στοιχείων που επιστρέφονται.

* **πεδίο callSequenceId, πεδίο callSequenceDate, εγγραφή σφάλματος** (βλέπε παράγραφο 5).

Όλα τα παραπάνω "εσωκλείονται" μαζί σε ένα JSON Object.

Ακολουθεί ένα παράδειγμα απόκρισης (JSON Response) για το API *getChildrenByParentAfm* της διαδικτυακής υπηρεσίας *familyInfoRestService*:

```
{
  "getChildrenByParentAfmOutputRecord": {
    "count": 2,
    "children": [
      {
        "amka": "150411*****",
        "name": "Δημήτριος",
        "lastname": "Παραδειγματόπουλος",
        "fathername": "Φανούριος",
        "mothername": "Κλεοπάτρα",
        "birthdate": "2011-04-15",
        "isDeceased": false
      },
      {
        "amka": "281212*****",
        "name": "Γεώργιος",
        "lastname": "Παραδειγματόπουλος",
        "fathername": "Φανούριος",
        "mothername": "Κλεοπάτρα",
        "birthdate": "2012-12-28",
        "isDeceased": false
      }
    ]
  },
  "callSequenceId": 11277554570,
  "callSequenceDate": "2024-08-27T16:42:07.852+03:00",
  "errorRecord": {}
}
```



9. Προαπαιτούμενα για κατανάλωση διαδικτυακών υπηρεσιών του ΚΕΔ

Η κατανάλωση διαδικτυακών υπηρεσιών του Κέντρου Διαλειτουργικότητας μπορεί να γίνει από οποιοδήποτε σύγχρονο περιβάλλον που υποστηρίζει δημιουργία συνδέσεων διαδικτύου (internet connections) μέσω πρωτοκόλλου TLSv1.2, αποστολή XML Requests (ή JSON Requests για τα REST Services) και λήψη XML Responses (ή JSON Responses για τα REST Services), ανεξάρτητα από το λειτουργικό σύστημα (Windows, Linux, MacOS κλπ) που χρησιμοποιείται.

Στα περισσότερα περιβάλλοντα θα χρειαστεί επιπλέον η λήψη και η εγκατάσταση του πιστοποιητικού σελίδας της ΓΓΠΣ (αρχείο gsis.gr.cer).

Ακολουθούν ενδεικτικά τα προπαιτούμενα σε γλώσσα προγραμματισμού Java και σε περιβάλλον ανάπτυξης ORACLE (PL/SQL) όπου έχουν γίνει δοκιμές στις διαδικτυακές υπηρεσίες του ΚΕΔ:

Java

- Java 1.8.x
- Λήψη πιστοποιητικού σελίδας gsis.gr (αρχείο gsis.gr.cer)
- Προσθήκη του πιστοποιητικού στην αποθήκη πιστοποιητικών της Java (java_home\jre\lib\security\cacerts)

Oracle (PL/SQL)

- Oracle 11g R2 ή ανώτερη
- Δυνατότητα χρήσης του πακέτου UTL_HTTP και ενεργοποίηση HTTPS πρόσβασης
- Λήψη πιστοποιητικού σελίδας gsis.gr (αρχείο gsis.gr.cer)
- Δημιουργία oracle wallet και προσθήκη του πιστοποιητικού



ΠΑΡΑΡΤΗΜΑ Α : Αναλυτική περιγραφή πεδίων Εγγραφής Ιχνηλασιμότητας

A/A	Περιγραφή	Τύπος Δεδομένων	Σχόλια
1	AUDIT_TRANSACTION_ID Μοναδικός A/A Αίτησης WS Φορέα / Επιχειρηματικής Μονάδας	VARCHAR2(100)	Μοναδικός κωδικός αίτησης ο οποίος δίνεται από την client εφαρμογή του Φορέα για τη συγκεκριμένη κλήση της υπηρεσίας. Με τη χρήση του κωδικού αυτού και μόνο ο φορέας θα πρέπει να είναι σε θέση να απαντήσει ποιος ήταν ο πραγματικός (τελικός) χρήστης της υπηρεσίας από την πλευρά του, καθώς και ποιες ανάγκες εξυπηρετούσε.
2	AUDIT_TRANSACTION_DATE Ημ/νία και Ώρα Αίτησης WS Φορέα / Επιχειρηματικής Μονάδας yyyy-mm-ddThh:mm:ssZ πχ 2016-10-10T10:08:24Z	DATE	Χρονοσήμανση Φορέα - Ημερομηνία και ώρα της κλήσης - από τον server του Φορέα που τρέχει την Client εφαρμογή και όχι το end user client pc.
3	AUDIT_UNIT Περιγραφή Φορέα / Επιχειρηματικής Μονάδας	VARCHAR2(100)	Το όνομα του Φορέα που καλεί τη διαδικτυακή υπηρεσία.
4	AUDIT_PROTOCOL Αρ. Πρωτοκόλλου Αίτησης WS	VARCHAR2(100)	Ενδέχεται η κλήση ενός web service να γίνεται από ένα Φορέα για να καλυφθούν συγκεκριμένες ανάγκες που σχετίζονται με μια Υπόθεση / Φάκελο που αναγνωρίζεται (από το Φορέα) με κάποιο μοναδικό τρόπο (π.χ. κάποιο αριθμό πρωτοκόλλου). Σ' αυτές τις περιπτώσεις, ο web service client θα πρέπει να υποστηρίζει τη δυνατότητα ο τελικός χρήστης να μπορεί να συμπληρώσει τις πληροφορίες αυτές.
5	AUDIT_USER_ID Αναγνωριστικό Χρήστη Φορέα / Επιχειρηματικής Μονάδας	VARCHAR2(100)	Το όνομα που χρησιμοποιεί ο τελικός χρήστης ώστε να συνδεθεί στην εφαρμογή πελάτη μέσω της οποίας γίνεται η συγκεκριμένη κλήση της διαδικτυακής υπηρεσίας. Το AUDIT_USER_ID μπορεί να είναι είτε κάποιο κωδικοποιημένο αναγνωριστικό είτε το πλήρες όνομα του τελικού χρήστη.
6	AUDIT_USER_IP Διεύθυνση IP του Τελικού Χρήστη Φορέα / Επιχειρηματικής Μονάδας	VARCHAR2(20)	Η διεύθυνση IP του τελικού χρήστη (end user pc) που κάνει χρήση της διαδικτυακής υπηρεσίας.



ΠΑΡΑΡΤΗΜΑ Β : Παραδείγματα κλήσης & απάντησης διαδικτυακών υπηρεσιών

SOAP SERVICES

1. XML Request διαδικτυακής λειτουργίας "getVehicleInformation" (Στοιχεία κατόχου οχήματος):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:veh="http://gsis.ggps.interoperability/VehicleInfoInterface">
  <soapenv:Header>
    <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken-1">
        <wsse:Username>T12345678T92345BY6KK4Y9E9MESVLRYY2W</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">*****</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <veh:getVehicleInformationRequest>
      <auditRecord>
        <auditTransactionId>1</auditTransactionId>
        <auditTransactionDate>2024-08-27T17:05:00Z</auditTransactionDate>
        <auditUnit>GSIS</auditUnit>
        <auditProtocol>1</auditProtocol>
        <auditUserId>user</auditUserId>
        <auditUserIp>0.0.0.0</auditUserIp>
      </auditRecord>
      <getVehicleInformationInputRecord>
        <arithmosKykloforias>NXY0000</arithmosKykloforias>
        <requestDate>2024-08-27</requestDate>
      </getVehicleInformationInputRecord>
    </veh:getVehicleInformationRequest>
  </soapenv:Body>
</soapenv:Envelope>
```



2. XML Response διαδικτυακής λειτουργίας "getVehicleInformation" (Στοιχεία κατόχου οχήματος):

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:getVehicleInformationResponse xmlns:ns2="http://gsis.ggps.interoperability/VehicleInfoInterface">
      <getVehicleInformationOutputRecord>
        <arithmosKykloforias>NXΥ0000</arithmosKykloforias>
        <marka>GORGOLIS S.A KRISS II</marka>
        <xrwma>ΧΡΥΣΑΦΙ</xrwma>
        <typosOxhmatos>ΔΙΚΥΚΛΟ</typosOxhmatos>
        <arithmosPlaisioy>XG8AN110HHGL*****</arithmosPlaisioy>
        <katastashOxhmatos>ΚΙΝΗΣΗ</katastashOxhmatos>
        <katoxoiList>
          <katoxos>
            <afm>000000000</afm>
            <percent>100</percent>
            <doy>1732</doy>
            <doyDesc>ΧΑΛΚΙΔΑΣ</doyDesc>
            <surname>ΠΑΡΑΔΕΙΓΜΑΤΟΠΟΥΛΟΣ</surname>
            <firstName>ΔΗΜΟΣΘΗΝΗΣ</firstName>
            <fathersFirstName>ΝΕΑΡΧΟΣ</fathersFirstName>
            <mothersFirstName>ΣΤΑΥΡΟΥΛΑ</mothersFirstName>
            <address>ΨΑΧΝΑ</address>
            <addressNo>0</addressNo>
            <parZipCode>34400</parZipCode>
            <parDescription>ΨΑΧΝΑ</parDescription>
          </katoxos>
        </katoxoiList>
      </getVehicleInformationOutputRecord>
      <callSequenceId>11277579800</callSequenceId>
      <callSequenceDate>2024-08-27T17:05:56.573+03:00</callSequenceDate>
      <errorRecord/>
    </ns2:getVehicleInformationResponse>
  </soap:Body>
</soap:Envelope>
```



REST SERVICES

1. JSON Request του REST API "validateID" (Επαλήθευση στοιχείων ταυτοτ. εγγράφου):

```
{
  "auditRecord": {
    "auditTransactionId": "1",
    "auditTransactionDate": "2024-08-27T17:14:53Z",
    "auditUnit": "GSIS",
    "auditProtocol": "1",
    "auditUserId": "user",
    "auditUserIp": "0.0.0.0"
  },
  "validateIDInputRecord": {
    "id": "AZ000000",
    "type": 1,
    "reason": "Δοκιμή"
  }
}
```

2. JSON Response του REST API "validateID" (Επαλήθευση στοιχείων ταυτοτ. εγγράφου):

```
{
  "validateIDOutputRecord": {
    "isFoundFlag": true,
    "isValidFlag": true,
    "idOut": "AZ000000",
    "issueDate": "28/04/2009",
    "issueInstitutionId": "1300",
    "issueInstitutionDesc": "Τ.Α. ΕΡΜΟΥΠΟΛΗΣ ΣΥΡΟΥ ΚΥΚΛΑΔΩΝ",
    "name": "ΔΑΜΙΑΝΟΣ",
    "surname": "ΠΑΡΑΔΕΙΓΜΑΤΟΠΟΥΛΟΣ",
    "fatherName": "ΠΗΛΕΑΣ",
    "motherName": "ΧΡΥΣΑΝΘΗ",
    "nameLatin": "DAMIANOS",
    "surnameLatin": "PARADIGMATOPOULOS",
    "fatherNameLatin": "PILEAS",
    "gender": "A",
    "birthDate": "24/01/1985",
    "birthPlace": "ΑΘΗΝΑ",
    "serviceCalled": "searchIdentitiesById",
    "idTypeDescr": "1 - ΑΤ ΑΣΤΥΝΟΜΙΚΗ ΤΑΥΤΟΤΗΤΑ"
  },
  "callSequenceId": 11277585294,
  "callSequenceDate": "2024-08-27T17:14:54.113+03:00",
  "errorRecord": {}
}
```