



G CONSELLERIA
O ADMINISTRACIONS
I PÚBLIQUES I
B MODERNITZACIÓ
/ DIRECCIÓ GENERAL
MODERNITZACIÓ I
ADMINISTRACIÓ DIGITAL

NOTIB: Manual d'integració amb Carpeta

octubre de 2020

Serveis d'Administració Electrònica en el Govern de les Illes Balears

Lot 2. Serveis d'administració electrònica



Control de versions del document

Control de Canvis			
Data	Autor	Versió	Canvis
23/10/20	Límit Technologies	1.0	Versió inicial

Revisat per		
Nom	Data	Àrea, departament o empresa

Aprovat per		
Nom	Data	Àrea, departament o empresa

Llista de distribució		
Nom	Àrea, departament o empresa	Correu electrònic



Índex

Control de versions del document	2
Objecte	4
Conceptes d'integració	4
Descripció del servei	4
Operacions	5
Model de dades	9
Rol necessari per a poder accedir al servei	12
Exemple integració	12
Integració REST	12
Dependències MAVEN	12
Client d'exemple v1	13
Exemple de petició REST	13
Annex I: Swagger amb informació de l'API REST de Carpeta	20



1. Objecte

En aquest document es proporcionen tots els requisits per preparar NOTIB per a la integració amb la carpeta mitjançant l'API REST.

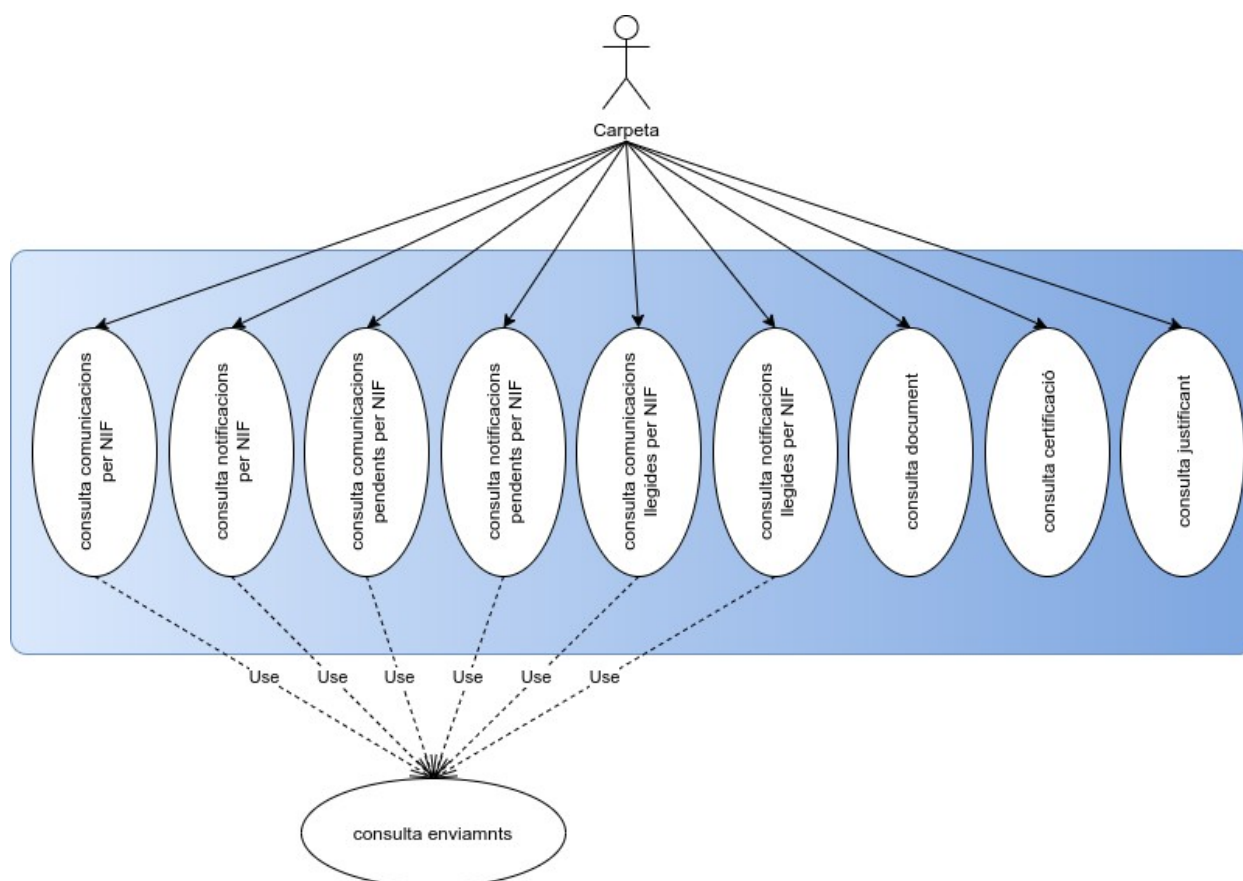
2. Conceptes d'integració

2.1. Descripció del servei

El servei de carpeta de NOTIB permet obtenir informació de les comunicacions i notificacions enviades a Notific@, a partir del DNI de l'interessat de la comunicació o notificació.

La API Rest proporciona informació de:

- les comunicacions o notificacions pendents per titular
- les comunicacions o notificacions realitzades per titular
- les comunicacions o notificacions per titular





2.1.1. Operacions

Les operacions que formen part d'aquest servei son els següents:

comunicacionsByTitular		
Tipus	GET	
Ruta	/api/consulta/v1/comunicacions/{dniTitular}	
Paràmetres		
dniTitular	String	DNI del titular de les comunicacions a consultar
Paràmetres tipus query de paginació (opcionals)		
Els paràmetres tipus query s'afegeixen a la petició a de la forma: http://url_peticio?param1=valor1¶m2=valor2		
pagina	int	Número de pàgina a mostrar en la paginació
mida	int	Mida de la pàgina a mostrar en la paginació
Resposta		
comunicacions	Resposta	Informació de totes les comunicacions d'un titular, i el seu estat
Descripcio	Consulta totes les comunicacions d'un titular donat el seu dni	

notificacionsByTitular		
Tipus	GET	
Ruta	/api/consulta/v1/notificacions/{dniTitular}	
Paràmetres		
dniTitular	String	DNI del titular de les notificacions a consultar
Paràmetres tipus query de paginació (opcionals)		
Els paràmetres tipus query s'afegeixen a la petició a de la forma: http://url_peticio?param1=valor1¶m2=valor2		
pagina	int	Número de pàgina a mostrar en la paginació
mida	int	Mida de la pàgina a mostrar en la paginació
Resposta		
comunicacions	Resposta	Informació de totes les notificacions d'un titular, i el seu estat
Descripcio	Consulta totes les comunicacions llegides d'un titular donat el seu dni	

comunicacionsPendantsByTitular		
Tipus	GET	
Ruta	/api/consulta/v1/comunicacions/{dniTitular}/pendents	
Paràmetres		
dniTitular	String	DNI del titular de les comunicacions a consultar
Paràmetres tipus query de paginació (opcionals)		
Els paràmetres tipus query s'afegeixen a la petició a de la forma: http://url_peticio?param1=valor1¶m2=valor2		
pagina	int	Número de pàgina a mostrar en la paginació
mida	int	Mida de la pàgina a mostrar en la paginació
Resposta		
comunicacions	Resposta	Informació sobre les comunicacions pendents d'un titular, i el seu estat
Descripcio	Consulta totes les comunicacions pendents (no llegides) d'un titular donat el seu dni	



Lot2. Serveis d'administració electrònica

notificacionsPendentsByTitular

Tipus	GET	
Ruta	/api/consulta/v1/notificacions/{dniTitular}/pendents	
Paràmetres		
dniTitular	String	DNI del titular de les comunicacions a consultar
Paràmetres tipus query de paginació (opcionals)		
Els paràmetres tipus query s'afegeixen a la petició a de la forma: http://url_peticio?param1=valor1¶m2=valor2		
pagina	int	Número de pàgina a mostrar en la paginació
mida	int	Mida de la pàgina a mostrar en la paginació
Resposta		
comunicacions	Resposta	Informació sobre les notificacions pendents d'un titular, i el seu estat
Descripció	Consulta totes les notificacions pendents (no llegides) d'un titular donat el seu dni	

comunicacionsLlegidesByTitular

Tipus	GET	
Ruta	/api/consulta/v1/comunicacions/{dniTitular}/llegides	
Paràmetres		
dniTitular	String	DNI del titular de les comunicacions a consultar
Paràmetres tipus query de paginació (opcionals)		
Els paràmetres tipus query s'afegeixen a la petició a de la forma: http://url_peticio?param1=valor1¶m2=valor2		
pagina	int	Número de pàgina a mostrar en la paginació
mida	int	Mida de la pàgina a mostrar en la paginació
Resposta		
comunicacions	Resposta	Retorna informació sobre les comunicacions ja llegides d'un titular, i el seu estat
Descripcio	Consulta totes les comunicacions llegides d'un titular donat el seu dni	

notificacionsLlegidesByTitular

Tipus	GET	
Ruta	/api/consulta/v1/notificacions/{dniTitular}/llegides	
Paràmetres		
dniTitular	String	DNI del titular de les comunicacions a consultar
Paràmetres tipus query de paginació (opcionals)		
Els paràmetres tipus query s'afegeixen a la petició a de la forma: http://url_peticio?param1=valor1¶m2=valor2		
pagina	int	Número de pàgina a mostrar en la paginació
mida	int	Mida de la pàgina a mostrar en la paginació
Resposta		
comunicacions	Resposta	Retorna informació sobre les notificacions ja llegides d'un titular, i el seu estat
Descripcio	Consulta totes les notificacions llegides d'un titular donat el seu dni	



Lot2. Serveis d'administració electrònica

getDocument		
Tipus	GET	
Ruta	/api/consulta/v1/document/{notificacioid}	
Paràmetres		
notificacioid	Long	Identificador de la comunicació/notificació a la que pertany el document a consultar
Resposta		
document	Arxiu	Document a consultar
Descripcio	Consulta un document donat l'identificador de la comunicació/notificació a la que pertany	

getCertificació		
Tipus	GET	
Ruta	/api/consulta/v1/certificacio/{enviamentId}	
Paràmetres		
enviamentId	Long	Identificador de l'enviamnet al que pertany la certificació a consultar
Resposta		
document	Arxiu	Document a consultar
Descripció	Obté la certificaciódonat l'identificador de l'enviament al que pertany	

getJustificant		
Tipus	GET	
Ruta	/api/consulta/v1/justificant/{enviamentId}	
Paràmetres		
enviamentId	Long	Identificador de l’enviamnet al que pertany el justificant a consultar
Resposta		
document	Arxiu	Document a consultar
Descripcio	Obté el justificant donat l’identificador de l’enviament al que pertany	

2.1.2. Model de dades

El model de dades que s'utilitzarà per les respostes és el següent:

Resposta			
Camp	Tipus	Card	Descripció
numeroElementsTotals	int	1	Nombre total d'elements
numeroElementsRetornats	int	1	Nombre d'elements retornat. Aquest valor dependrà de la paginació
resultat	Transmissió	0..n	Llistat de les comunicacions/notificacions



Lot2. Serveis d'administració electrònica

Transmissió			
Camp	Tipus	Card	Descripció
id	Long	1	Identificador de la comunicació
emisor	String	1	Codi dir3 de l'entitat emisora
organGestor	String	1	Codi dir3 de l'òrgan gestor
procediment	String	0..1	Codi SIA del procediment
numExpedient	String	0..1	Número de l'expedient al que està associada la comunicació
concepte	String	1	Concepte de la comunicació
descripcio	String	0..1	Descripció de la comunicació
dataEnviament	Date	1	Data d'enviament de la comunicació
estat	Estat	1	Estat de l'enviament
dataEstat	Date	1	Data en que s'ha canviar l'estat de l'enviament
document	Document	1	Document comunicat
titular	Persona	1	Persona titular de l'enviament
destinatari	Persona	0..n	Persones representants, destinatàries de l'enviament
subestat	SubEstat	0..1	Subestat de l'enviament
dataSubestat	Date	0..1	Data en que s'ha canviat al subestat actual
error	boolean	1	Informa si s'ha produït algun error en la comunicació
errorData	Date	0..1	Data de l'error
errorDescripcio	String	0..1	Descripció de l'error
justificant	String	0..1	Url per a obtenir el justificant de la comunicació
certificacio	String	0..1	Url per a obtenir la certificació de compareixença de la notificació

Document			
Camp	Tipus	Card	Descripció
nom	Long	1	Nom del document, inclosa l'extensió d'aquest
mediaType	String	0..1	Tipus mime del document
mida	Integer	0..1	Mida en bytes del contingut del document
url	String	1	Url del mètode de la API Rest per a obtenir el document

Persona			
Camp	Tipus	Card	Descripció
nom	String	1	Nom de la persona. Correspon a la raó social quan no es tracta d'una persona física
l·linatge1	String	0..1	Primer l·linatge de la persona física
l·linatge2	String	0..1	Segon l·linatge de la persona física
nif	String	1	document identificatiu de la persona
email	String	0..1	Correu electrònic de la persona
tipus	PersonaTipus	1	Tipus de persona



Lot2. Serveis d'administració electrònica

Estat

Enumerat amb els següents valors possibles:

PENDENT	Comunicació / Notificació pendent d'enviar
REGISTRADA	Comunicació / Notificació registrada, però pendent d'enviar a Notifica
ENVIADA	Comunicació / Notificació enviada a notificà
FINALITZADA	Comunicació / Notificació finalitzada (es troba en un estat final: llegida, rebutjada, ...)
PROCESSADA	Comunicació / Notificació finalitzada i marcada com a processada

SubEstat

Enumerat amb els següents valors possibles:

NOTIB_PENDENT	La notificació està pendent d'enviar a Notifica
NOTIB_ENVIADA	La notificació s'ha enviat correctament a Notifica
ABSENT	Només aplica a enviaments postals i l'empleat de l'Operador Postal ho pot indicar de forma explícita
ADRESA_INCORRECTA	Només aplica a enviaments postals i l'empleat de l'Operador Postal ho pot indicar de forma explícita
DESCONEGUT	Només aplica a enviaments postals i l'empleat de l'Operador Postal ho pot indicar de forma explícita
ENVIADA_CI	Enviat a CIE
ENVIADA_DEH	Enviat a DEH
ENVIAMENT_PROGRAMA T	La notificació es troba en espera de ser enviada en una data futura indicada per l'usuari
ENTREGADA_OP	S'ha entregat a l'Operador Postal
ERROR_ENTREGA	S'ha produït un error en l'entrega a l'Agent Colaborador
EXPIRADA	Ha passar el plaç establert sense poder ser entregada
EXTRAVIADA	Només per enviaments postals. Ho retorna l'Operador Postal
MORT	El destinatari de l'enviament ha mort
LLEGIDA	Comunicació llegida
NOTIFICADA	Notificació llegida
PENDENT	Pendent de registrar i enviar a Notifica
PENDENT_ENVIAMENT	Pendedent d'enviar a CIE o DEH, segons correspongui
PENDENT_SEU	Pendent de lectura a la seu, abans de ser enviada a CIE o DEH
PENDENT_CIE	L'enviament s'ha d'entregar al CIE
PENDENT_DEH	L'enviament s'ha d'entrega a la DEH
REBUTJADA	Rebutjada per l'interessat
SENSE_INFORMACIO	
FINALITZADA	Notifica ha retornat un estat final
ENVIADA	Enviada a Notifica, però pendent de finalitzar
REGISTRADA	Registrada i pendent d'enviar a Notifica
PROCESSADA	Finalitzada i marcada com a processada
ANULADA	Notificació anulada, al ser errònea
ENVIAT_SIR	Comunicació enviada via SIR



PersonaTipus	
Enumerat amb els següents valors possibles:	
ADMINISTRACIO	Es tracta d'una administració
FISICA	Es tracta d'una persona física
JURIDICA	Es tracta d'una persona jurídica

Arxiu			
Camp	Tipus	Card	Descripció
nom	Long	1	Nom del document, inclosa l'extensió d'aquest
mediaType	String	0..1	Tipus mime del document
contingut	String	1	Contingut del document codificat en Base64

2.2. Rol necessari per a poder accedir al servei

Per a poder accedir al servei és necessari disposar d'un usuari amb el rol NOT_CARPETA. En cas contrari es rebrà un error 403 de recurs prohibit.

3. Exemple integració

La integració amb l'aplicació Notib es realitzarà mitjançant el servei web REST

3.1. Integració REST

3.1.1. Dependències MAVEN

Per a desenvolupar un client per al servei de carpeta es pot utilitzar una llibreria com jersey. El més senzill és incloure el següent bloc a dins el fitxer pom.xml del projecte maven:

Per a jboss 5.2, a dins la secció <dependencies>:

```
<dependency>
  <groupId>com.sun.jersey</groupId>
  <artifactId>jersey-client</artifactId>
  <version>1.19.3</version>
</dependency>
```



3.2. Client d'exemple v1

3.2.1. Exemple de petició REST

La següent classe es correspon a un joc de proves que utilitza una classe client per a realitzar les peticions cap a Notib:

```
import static org.junit.Assert.assertNotNull;
import static org.junit.Assert.fail;
import java.io.IOException;
import javax.xml.datatype.DatatypeConfigurationException;
import org.apache.commons.codec.DecoderException;
import org.junit.Before;
import org.junit.Test;
import es.caib.notib.core.api.rest.consulta.Arxiu;
import es.caib.notib.core.api.rest.consulta.Resposta;

/**
 * Test per al client REST del servei de carpeta de NOTIB.
 *
 * @author Limit Technologies <limit@limit.es>
 */
public class CarpetaRestTest {

    private static final String URL = "http://localhost:8080/notib";
    private static final String USERNAME = "notibcar";
    private static final String PASSWORD = "notibcar";

    private CarpetaRestClient client;

    @Before
    public void setUp() throws IOException, DecoderException {
        client = new CarpetaRestClient(
            URL,
            USERNAME,
            PASSWORD,
            true);
    }

    @Test
    public void testConsulta() throws DatatypeConfigurationException, IOException,
    DecoderException {
        String nif = "0000000t";
        int pagina = 0;
        int mida = 10;

        try {
            Resposta resposta = client.consultaNotificacions(
                nif,
                pagina,
                mida);

            assertNotNull(resposta);
            assertNotNull(resposta.getNumeroElementsTotals());
            assertNotNull(resposta.getNumeroElementsRetornats());
        }
    }
}
```



```
        } catch (Exception e) {  
            fail("Error en la petició");  
        }  
    }  
  
    @Test  
    public void testDocument() throws DatatypeConfigurationException, IOException,  
DecoderException {  
        Long notificacioId = 173875L;  
  
        try {  
            Arxiu document = client.consultaDocument(notificacioId);  
  
            assertNotNull(document);  
            assertNotNull(document.getNom());  
            assertNotNull(document.getContingut());  
        } catch (Exception e) {  
            fail("Error en la petició");  
        }  
    }  
}
```

La següent classe es correspon al client per a accedir a Notib. En aquest exemple s'ha implementat la creació d'un clientjersey, amb autenticació, i la implementació de don mètodes, un per a obtenir notificación, i l'altre per a obtenir documents.

La implementació de la resta d'operacions disponibles a Notib s'implementarien exactament igual que les de l'exemple:

```
import java.rmi.RemoteException;  
import java.util.ArrayList;  
  
import javax.ejb.CreateException;  
import javax.management.InstanceNotFoundException;  
import javax.management.MalformedObjectNameException;  
import javax.naming.NamingException;  
import javax.ws.rs.core.Response;  
import javax.ws.rs.core.UriBuilder;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
import com.fasterxml.jackson.databind.ObjectMapper;  
import com.sun.jersey.api.client.Client;  
import com.sun.jersey.api.client.ClientHandler;  
import com.sun.jersey.api.client.ClientHandlerException;  
import com.sun.jersey.api.client.ClientRequest;  
import com.sun.jersey.api.client.ClientResponse;  
import com.sun.jersey.api.client.filter.ClientFilter;  
import com.sun.jersey.api.client.filter.HTTPBasicAuthFilter;  
import com.sun.jersey.api.representation.Form;  
  
import es.caib.notib.core.api.rest.consulta.Arxiu;  
import es.caib.notib.core.api.rest.consulta.Resposta;;
```



```
/**
 * Client REST per al servei de carpeta de NOTIB.
 *
 * @author Limit Technologies <limit@limit.es>
 */
public class CarpetaRestClient {

    private static final String CARPETA_SERVICE_PATH = "/api/consulta/v1";

    private String baseUrl;
    private String username;
    private String password;

    private boolean autenticacioBasic = false;

    public CarpetaRestClient() {}
    public CarpetaRestClient(
        String baseUrl,
        String username,
        String password) {
        super();
        this.baseUrl = baseUrl;
        this.username = username;
        this.password = password;
    }

    public CarpetaRestClient(
        String baseUrl,
        String username,
        String password,
        boolean autenticacioBasic) {
        super();
        this.baseUrl = baseUrl;
        this.username = username;
        this.password = password;
        this.autenticacioBasic = autenticacioBasic;
    }

    public Resposta consultaNotificacions(
        String nif,
        Integer pagina,
        Integer mida) {
        try {
            String urlAmbMetode = baseUrl + CARPETA_SERVICE_PATH + "/notificacions/" + nif;
            String paginacio = "";
            if (pagina != null && mida != null) {
                paginacio = "?pagina=" + pagina + "&mida=" + mida;
            }
            Client jerseyClient = generarClient();
            if (username != null) {
                autenticarClient(
                    jerseyClient,
                    urlAmbMetode,
                    username,
                    password);
            }
            String json = jerseyClient.
                resource(urlAmbMetode + paginacio).
                type("application/json").

```



```
        get(String.class);
        ObjectMapper mapper = new ObjectMapper();
        return mapper.readValue(json, Resposta.class);
    } catch (Exception ex) {
        throw new RuntimeException(ex);
    }
}

public Arxiu consultaDocument(
    Long notificacioId) {
    try {
        String urlAmbMetode = baseUrl + CARPETA_SERVICE_PATH + "/document/" +
notificacioId;
        Client jerseyClient = generarClient();
        if (username != null) {
            autenticarClient(
                jerseyClient,
                urlAmbMetode,
                username,
                password);
        }
        String json = jerseyClient.
            resource(urlAmbMetode).
            type("application/json").
            get(String.class);
        ObjectMapper mapper = new ObjectMapper();
        return mapper.readValue(json, Arxiu.class);
    } catch (Exception ex) {
        throw new RuntimeException(ex);
    }
}

public boolean isAutenticacioBasic() {
    return autenticacioBasic;
}

private Client generarClient() {
    Client jerseyClient = Client.create();
    jerseyClient.addFilter(
        new ClientFilter() {
            private ArrayList<Object> cookies;
            @Override
            public ClientResponse handle(ClientRequest request) throws
ClientHandlerException {
                if (cookies != null) {
                    request.getHeaders().put("Cookie", cookies);
                }
                ClientResponse response = getNext().handle(request);
                if (response.getCookies() != null) {
                    if (cookies == null) {
                        cookies = new ArrayList<Object>();
                    }
                    cookies.addAll(response.getCookies());
                }
                return response;
            }
        }
    );
    jerseyClient.addFilter(
```



```

        new ClientFilter() {
            @Override
            public ClientResponse handle(ClientRequest request) throws
ClientHandlerException {
                ClientHandler ch = getNext();
                ClientResponse resp = ch.handle(request);

                if (resp.getStatusInfo().getFamily() !=
Response.Status.Family.REIRECTION) {
                    return resp;
                } else {
                    String redirectTarget =
resp.getHeaders().getFirst("Location");
                    request.setURI(UriBuilder.fromUri(redirectTarget).build());
                    return ch.handle(request);
                }
            }
        };
        return jerseyClient;
    }

    private void autenticarClient(
        Client jerseyClient,
        String urlAmbMetode,
        String username,
        String password) throws InstanceNotFoundException, MalformedURLException,
RemoteException, NamingException, CreateException {
        if (!autenticacioBasic) {
            logger.debug(
a damunt jBoss (" +
                "urlAmbMetode=" + urlAmbMetode + ", " +
                "username=" + username +
                "password=*****");
            jerseyClient.resource(urlAmbMetode).get(String.class);
            Form form = new Form();
            form.putSingle("j_username", username);
            form.putSingle("j_password", password);
            jerseyClient.
            resource(baseUrl + "/j_security_check").
            type("application/x-www-form-urlencoded").
            post(form);
        } else {
            logger.debug(
                "Autenticant REST amb autenticació de tipus HTTP basic (" +
                "urlAmbMetode=" + urlAmbMetode + ", " +
                "username=" + username +
                "password=*****");
            jerseyClient.addFilter(
                new HTTPBasicAuthFilter(username, password));
        }
    }

    private static final Logger logger = LoggerFactory.getLogger(CarpetaRestClient.class);
}

```



A continuació es mostren les classes del model de dades. Aquestes classes utilitzen la llibreria lombok per a simplificar-les.

```
import java.util.List;
import lombok.Data;
import lombok.NoArgsConstructor;
```

```
@Data
@NoArgsConstructor
public class Resposta {
```

```
    int numeroElementsTotals;
    int numeroElementsRetornats;
    List<Transmissio> resultat;
```

```
}
```

```
import java.util.Date;
import java.util.List;
import lombok.Data;
```

```
/**
 * Informació d'una Transmissió (Notificació / Comunicació)
 * @author Limit Technologies <limit@limit.es>
 */
```

```
@Data
public class Transmissio {
```

```
    // Notificació
    private Long id;
    private String emisor;
    private String organGestor;
    private String procediment;
    private String numExpedient;
    private String concepte;
    private String descripcio;
    private Date dataEnviament;
    private Estat estat;
    private Date dataEstat;
    private Document document;
```

```
    // Enviament
    private Persona titular;
    private List<Persona> destinataris;
    private SubEstat subestat;
    private Date dataSubestat;
```

```
    // Error
    private boolean error;
    private Date errorData;
    private String errorDescripcio;
```

```
    private String justificant;
    private String certificacio;
```

```
}
```




```
import lombok.Data;  
import lombok.NoArgsConstructor;
```

```
@Data  
@NoArgsConstructor  
public class Persona {  
  
    private PersonaTipus tipus;  
    private String nom;  
    private String llinatge1;  
    private String llinatge2;  
    private String nif;  
    private String email;  
  
}
```

```
import lombok.AllArgsConstructor;  
import lombok.Builder;  
import lombok.Data;  
import lombok.NoArgsConstructor;
```

```
@Data  
@Builder  
@NoArgsConstructor @AllArgsConstructor  
public class Document {  
  
    private String nom;  
    private String mediaType;  
    private Long mida;  
    private String url;  
  
}
```

```
@Data  
@Builder  
@NoArgsConstructor @AllArgsConstructor  
public class Arxiu {  
  
    String nom;  
    String mediaType;  
    String contingut;  
  
}
```

```
public enum Estat {  
    PENDENT,  
    ENVIADA,  
    REGISTRADA,  
    FINALITZADA,  
    PROCESSADA;  
}
```



```
public enum SubEstat {  
    NOTIB_PENDENT,  
    NOTIB_ENVIADA,  
    ABSENT,  
    ADRESA_INCORRECTA,  
    DESCONEGUT,  
    ENVIADA_CI,  
    ENVIADA_DEH,  
    ENVIAMENT_PROGRAMAT,  
    ENTREGADA_OP,  
    ERROR_ENTREGA,  
    EXPIRADA,  
    EXTRAVIADA,  
    MORT,  
    LLEGIDA,  
    NOTIFICADA,  
    PENDENT,  
    PENDENT_ENVIAMENT,  
    PENDENT_SEU,  
    PENDENT_CIE,  
    PENDENT_DEH,  
    REBUTJADA,  
    SENSE_INFORMACIO,  
    FINALITZADA,  
    ENVIADA,  
    REGISTRADA,  
    PROCESSADA,  
    ANULADA,  
    ENVIAT_SIR  
}
```

```
public enum PersonaTipus {  
    ADMINISTRACIO,  
    FISICA,  
    JURIDICA;  
}
```

Annex I: Swagger amb informació de l'API REST de Carpeta

A l'adreça http://URL_NOTIB/notib/api/rest es disposa d'informació de la API REST de la Carpeta, proporcionada mitjançant l'eina Swagger.

Aquesta eina mostra els paràmetres que ha de rebre cada operació, i es que retorna, així com exemple dels json utilitzats.



Lot2. Serveis d'administració electrònica

api/consulta/v1

Mostrar/Ocultar | Llistar Operacions | Expandir Operacions

GET	/api/consulta/v1/certificacio/{enviamentId}	Obté la certificació d'una notificació
GET	/api/consulta/v1/comunicacions/{dniTitular}	Consulta totes les comunicacions d'un titular donat el seu dni
GET	/api/consulta/v1/comunicacions/{dniTitular}/llegides	Consulta totes les comunicacions llegides d'un titular donat el seu dni
GET	/api/consulta/v1/comunicacions/{dniTitular}/pendents	Consulta totes les comunicacions pendents (no llegides) d'un titular donat el seu dni
GET	/api/consulta/v1/document/{notificacioid}	Obté el document d'una notificació
GET	/api/consulta/v1/justificant/{enviamentId}	Obté el justificant d'una comunicació
GET	/api/consulta/v1/notificacions/{dniTitular}	Consulta totes les notificacions d'un titular donat el seu dni
GET	/api/consulta/v1/notificacions/{dniTitular}/llegides	Consulta totes les notificacions llegides d'un titular donat el seu dni
GET	/api/consulta/v1/notificacions/{dniTitular}/pendents	Consulta totes les notificacions pendents (no llegides) d'un titular donat el seu dni

Notes de implementació

Retorna Informació sobre les notificacions pendents d'un titular, i el seu estat

Classe de la Resposta (Status 200)

OK

Model | Valor d'Exemple

```
{
  "numeroElementsTotals": 0,
  "numeroElementsRetornats": 0,
  "resultat": [
    {
      "id": 0,
      "emisor": "string",
      "organGestor": "string",
      "procediment": "string",
      "numExpedient": "string"
    }
  ]
}
```

Tipus de Contingut (Content Type) de la Resposta application/json

Paràmetres

Paràmetre	Valor	Descripció	Tipus del Paràmetre	Tipus de la Dada
dniTitular	<input type="text" value="(required)"/>	DNI del titular de les comunicacions a consultar	path	string
pagina	<input type="text"/>	Número de pàgina a mostrar en la paginació	query	integer
mida	<input type="text"/>	Mida de la pàgina a mostrar en la paginació	query	integer

Missatges de la Resposta

Codi de Status HTTP	Raó	Model de la Resposta	Headers
401	Unauthorized		