

# NLL FHIR Connectathon 9 februari 2023

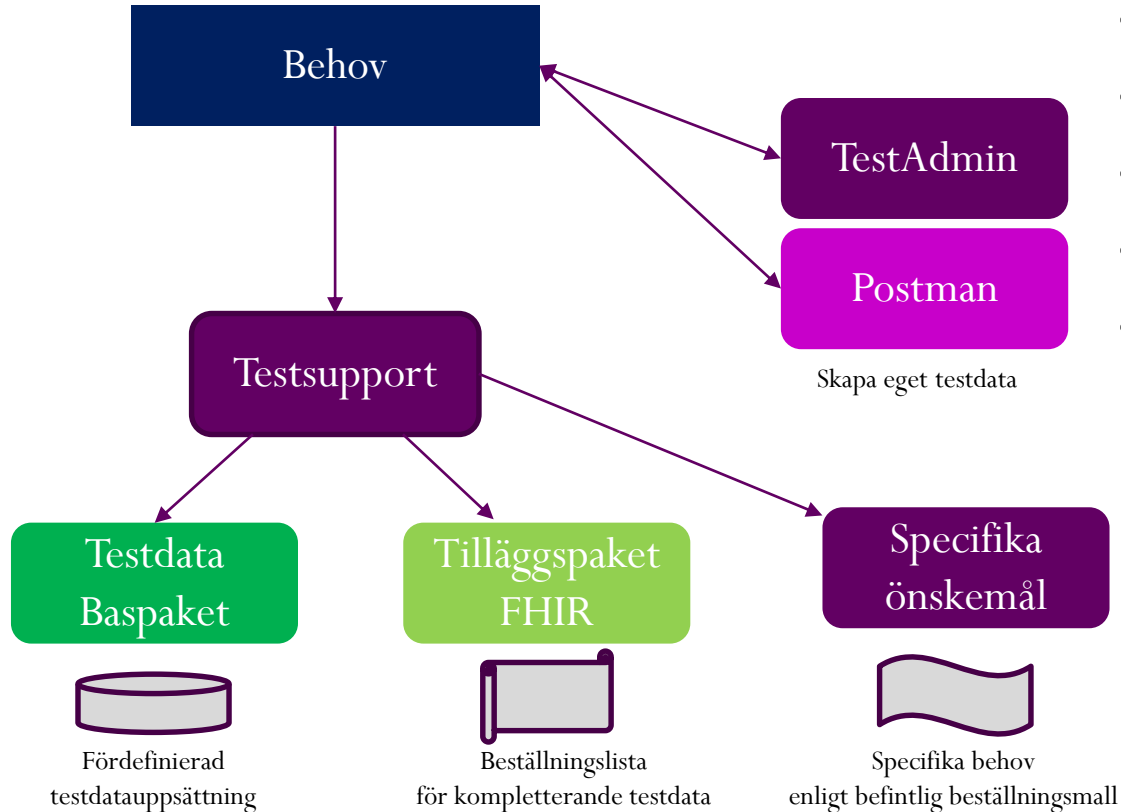
Välkommen!



# Agenda

- **Inledning**
- Genomgång av strukturen i Postman
- Praktiska övningar
  - Förberedelser
  - Köra några av användningsfallen
  - Skapa nya användningsfall
  - Egna labbar
- Avslutning

# Bakgrund



- Informationsmöte – testdata
- Demo Postmanprojekt
- Önskemål om fördjupning
- Dagens Connectathon.
- Presentation utskickad

# Mål

Att informera om och genomföra övningar med det Postmanprojekt som skapats i syfte att det skall bli ett användbart verktyg för systemleverantörer.

# Syfte

- Förstå hur postmanprojektet fungerar
- Förstå hur man kan modifiera befintliga användningsfall baserat på befintliga användningsfall
- Förstå hur man kan skapa helt nya användningsfall baserat på befintliga användningsfall

# Målgrupp

Primär målgrupp är arkitekter, systemutvecklare och testare som arbetar med utveckling av vård- och apotekssystem som ska integrera med NLL.

# Innan Connectathon

## Importera Postmanfiler

- Postmanfiler har mejlats till inbjudna personer. Om du inte fått filerna, kontakta [staffan.olsson@ehalsomyndigheten.se](mailto:staffan.olsson@ehalsomyndigheten.se).
- Filerna ska importeras i Postman:
- Installera Postman från [Postman API Platform](#).
- Packa upp innehållet i zip-filen som mejlats ut.
- Importera samtliga json-filer i zip-filen till Postman.

# Innan Connectathon

## Klientcertifikat

- För att komma åt Nationella läkemedelslistan krävs anslutning med Mutual-TLS. För connectathonet används testcertifikat från Expisoft (Testcertifikat Server-organisation och Stämpellegitimationer, Bolag A).
- För att ladda ned testcertifikat för Bolag A gå till <https://eid.expisoft.se/hamta-e-legitimation/> och ange identifierare 2D38-5C67-DA63-8AB8-T1A1
- I Postman gå till "File/Settings/Certificates" och klicka på "Add Certificate".

# Innan Connectathon

## Klientcertifikat, fortsättning

- I "Host" ange "\*.extps.ecp.ehalsomyndigheten.se"
- I "PFX file" ladda upp den certifikatfilen som tagits hem från Expisoft, xxx.p12
- I "Passphrase" ange "4468005265525061"
- Klicka på "Add"
- Lägg till ytterligare ett certifikat på samma sätt (steg 2-6) med skillnaden att i "Host" (steg 3) anges "\*.extp.ecp.ehalsomyndigheten.se"

# Innan Connectathon

## Konnektivitetstest

- Inför mötet ska deltagarna göra ett konnektivitetstest, dvs säkerställa att Postmanklienten har åtkomst till testmiljön (EXT18).
- Efter att ni importerat Postmanfiler och lagt till klientcertifikat, säkerställ att Ext18 är valt i drop down uppe i högra hörnet (val av "Environment")
- Öppna de importerade json-filerna och kör "1150 Anslutningstest/ GET \$ping"
- Ett lyckat anrop ska få ett svar med http-status 200.

# Innan Connectathon

## Konnektivitetstest

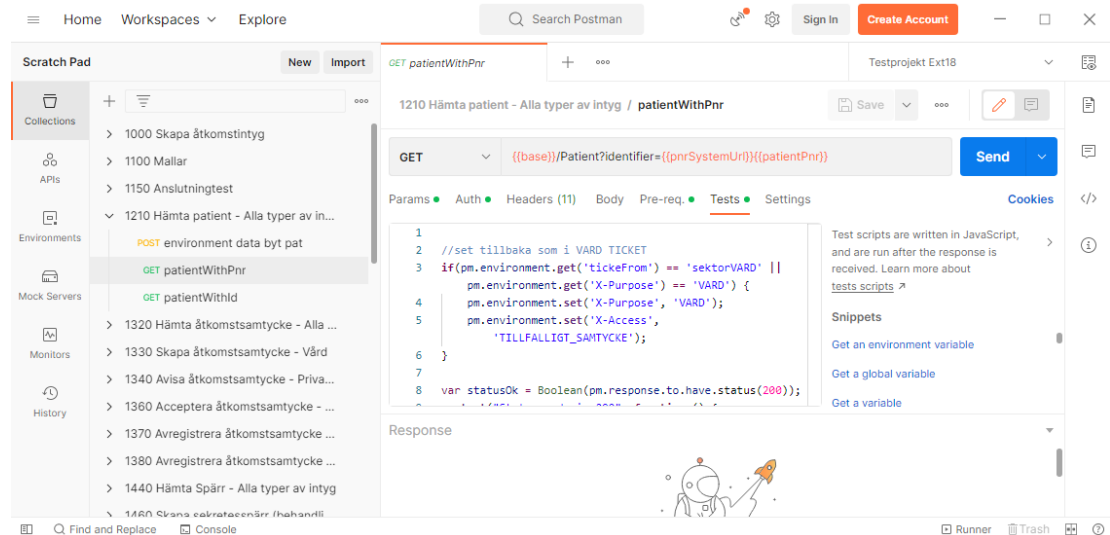
- Vi önskar att ni återkopplar resultatet till [staffan.olsson@ehalsomyndigheten.se](mailto:staffan.olsson@ehalsomyndigheten.se) **senast den 7 februari**. Om ni har problem, kontakta Staffan.
- OBS. E-hälsomyndigheten erbjuder ingen support kring själva verktyget Postman, enbart support kring tjänsteanropen, klientcertifikat o.d.
- Mer information om hur man använder Postman finns på [Learning Center | Postman Learning Center](#).

# Agenda

- Inledning
- **Genomgång av strukturen i Postman**
- Praktiska övningar
  - Förberedelser
  - Köra några av användningsfallen
  - Skapa nya användningsfall
  - Egna labbar
- Avslutning

# Genomgång av principer

- Det här avsnittet beskriver hur Postman har används för att konstruera de olika användningsfallen.
- Det är inte en komplett beskrivning av hur samtliga användningsfall är konstruerade.



The screenshot displays the Postman application interface. The top navigation bar includes 'Home', 'Workspaces', and 'Explore'. A search bar for 'Search Postman' is present. On the right, there are 'Sign In' and 'Create Account' buttons. The main workspace shows a REST client request for '1210 Hämta patient - Alla typer av intyg / patientWithPnr'. The request is a GET method with the URL template: `{{(base)}}/Patient?identifier={{(pnrSystemUri)}}{{(patientPnr)}}`. The 'Tests' tab is active, showing JavaScript code for environment variable management and status checks. The code includes comments in Swedish and uses Postman's `pm.environment` and `pm.response` objects. A 'Response' section is visible at the bottom, and a small cartoon character is in the bottom right corner.

```
1 //set tillbaka som i VARD TICKET
2
3 if(pm.environment.get('tickeFrom') == 'sektorVARD' ||
4   pm.environment.get('X-Purpose') == 'VARD') {
5   pm.environment.set('X-Purpose', 'VARD');
6   pm.environment.set('X-Access',
7     'TILLFALLIGT_SAMTYCKE');
8 }
9
10 var statusOk = Boolean(pm.response.to.have.status(200));
```

# Globala variabler

Globals
+ ...
Ext1

**Globals**

Global variables for a workspace are a set of variables that are always available within the scope of that workspace. They are available to anyone in that workspace. [Learn more about globals](#) ↗

	VARIABLE	TYPE ⓘ	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ
<input checked="" type="checkbox"/>	pnrSystemUrl	default ▾	http%3A%2F%2Felectronich...	http%3A%2F%2Felectro
<input checked="" type="checkbox"/>	X-Org-Info-encoded	default ▾		
<input checked="" type="checkbox"/>	X-User-Agent-encoded	default ▾		
<input checked="" type="checkbox"/>	X-User-Agent	default ▾		
<input checked="" type="checkbox"/>	X-Org-Info	default ▾		
<input checked="" type="checkbox"/>	URNOIDForskrivarKod	default ▾	urn%3Aoid%3A1.2.752.116.3.1...	urn%3Aoid%3A1.2.752.1
<input checked="" type="checkbox"/>	URNOIDLegitimationKod	default ▾	urn%3Aoid%3A1.2.752.116.3.1...	urn%3Aoid%3A1.2.752.1
<input checked="" type="checkbox"/>	currentDate	default ▾		

De globala variabler som saknar värde i bilden uppdateras av användningsfallen.

De variabler som har värden i bilden används som konstanter.

M.a.o. inget ska ändras här.

# Miljövariabler

Testprojekt Ext18 Fork

	VARIABLE	TYPE ⓘ	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ
<input checked="" type="checkbox"/>	base	default ▾	https://api-ext18-ecp1.extps....	https://api-ext18-ecp1.ex
<input checked="" type="checkbox"/>	webTicket	default ▾	https://webticketserver-ext18...	https://webticketserver-e
<input checked="" type="checkbox"/>	nplPackId	default ▾	19860401100253	19860401100253
<input checked="" type="checkbox"/>	nplId	default ▾	19860613000068	19860613000068
<input checked="" type="checkbox"/>	X-Access	default ▾		
<input checked="" type="checkbox"/>	X-Purpose	default ▾		
<input checked="" type="checkbox"/>	patientPnr	default ▾		
<input checked="" type="checkbox"/>	patientId	default ▾		

Tips!

Skapa en kopia av Ext18 om du vill ansluta till en annan miljö.

Adress till FHIR-tjänsten

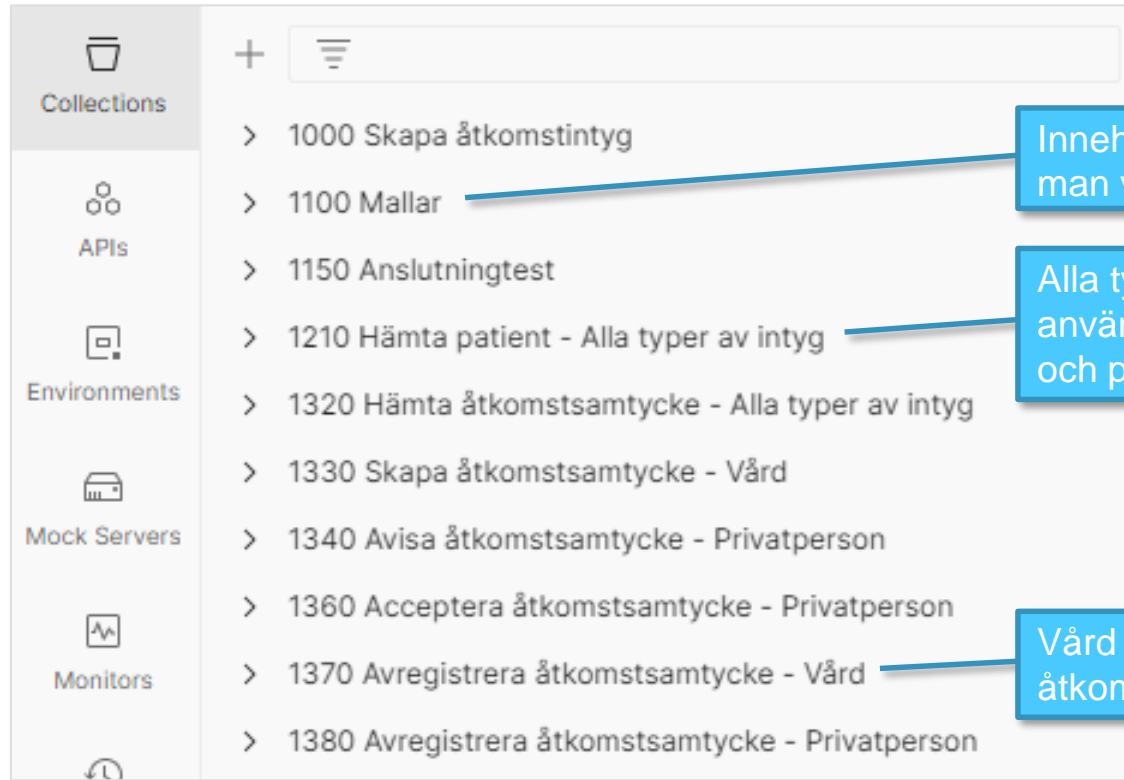
Adress till intygstjänsten

NPL-PackId och NPL-Id ändras här om man vill använda ett annat läkemedel.

patientPnr kan anges här eller via skript

Övriga variabler som saknar värde uppdateras av användningsfallen och ska inte anges här.

# Användningsfall



The screenshot shows a sidebar with categories: Collections, APIs, Environments, Mock Servers, and Monitors. The main area displays a list of use cases with expandable arrows. Three blue callout boxes point to specific items in the list.






Category	Use Case
Collections	> 1000 Skapa åtkomstintyg
APIs	> 1100 Mallar
APIs	> 1150 Anslutningstest
Environments	> 1210 Hämta patient - Alla typer av intyg
Environments	> 1320 Hämta åtkomstsamtycke - Alla typer av intyg
Mock Servers	> 1330 Skapa åtkomstsamtycke - Vård
Mock Servers	> 1340 Avisa åtkomstsamtycke - Privatperson
Mock Servers	> 1360 Acceptera åtkomstsamtycke - Privatperson
Monitors	> 1370 Avregistrera åtkomstsamtycke - Vård
Monitors	> 1380 Avregistrera åtkomstsamtycke - Privatperson

Innehåller mallar som kan användas om man vill skapa egna användningsfall

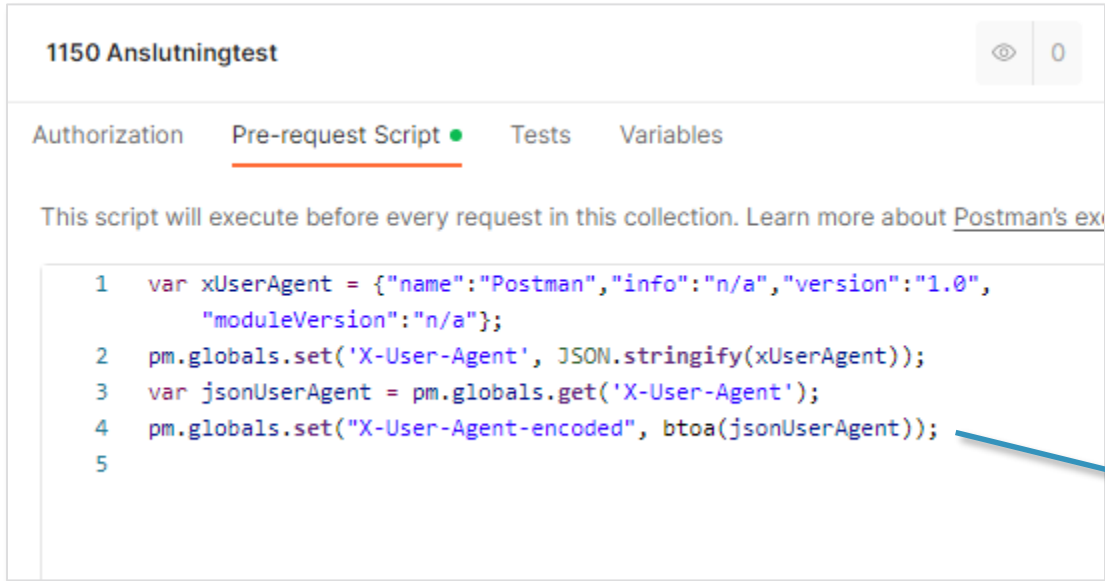
Alla typer betyder att det går att använda åtkomstintyg för vård, apotek och privatperson

Vård anger att det går att använda åtkomstintyg för vård

# Användningsfall forts.

Collections	+ 
 APIs	<ul style="list-style-type: none"><li data-bbox="357 333 937 365">&gt; 1440 Hämta Spärr - Alla typer av intyg</li><li data-bbox="357 404 1226 436">&gt; 1460 Skapa sekretesspärr (behandlingsorsak SEKRETESS...</li><li data-bbox="357 475 1226 507">&gt; 1462 Häva sekretesspärr (behandlingsorsak SEKRETESS_...</li></ul>
 Environments	<ul style="list-style-type: none"><li data-bbox="357 540 1226 573">&gt; 1464 Skapa integretesspärr (behandlingsorsak INTEGRET...</li><li data-bbox="357 611 1226 644">&gt; 1466 Häva integretetsspärr (behandlingsorsak INTEGRITE...</li></ul>
 Mock Servers	<ul style="list-style-type: none"><li data-bbox="357 677 1014 709">&gt; 1710 Hämta förskrivning - Alla typer av intyg</li><li data-bbox="357 748 840 780">&gt; 1720 Skapa förskrivning - Vård</li></ul>
 Monitors	<ul style="list-style-type: none"><li data-bbox="357 813 937 846">&gt; 1810 Hämta uttag - Alla typer av intyg</li><li data-bbox="357 884 782 917">&gt; 1820 Skapa uttag - Apotek</li></ul>

# Skript på Collection-nivå



```
1150 Anslutningstest 👁 0
```

Authorization Pre-request Script Tests Variables

This script will execute before every request in this collection. Learn more about [Postman's ex](#)

```
1 var xUserAgent = {"name":"Postman","info":"n/a","version":"1.0",
2   "moduleVersion":"n/a"};
3 pm.globals.set('X-User-Agent', JSON.stringify(xUserAgent));
4 var jsonUserAgent = pm.globals.get('X-User-Agent');
5 pm.globals.set("X-User-Agent-encoded", btoa(jsonUserAgent));
```

I en Collection kan man skapa skript som körs för samtliga ingående användningsfall.

Denna teknik används bara i 1150 – Anslutningstest.

x-user-agent sätts när man tar ut ett åtkomstintyg men \$ping kräver inget åtkomstintyg.

# Anslutningstest – indata

1150 Anslutningstest / GET Sping

GET `{{base}}/Sping`

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Headers 6 hidden

	KEY	VALUE
<input checked="" type="checkbox"/>	x-request-id	<code>{{ \$guid }}</code>
<input checked="" type="checkbox"/>	x-user-agent	<code>{{ X-User-Agent-encoded }}</code>
	Key	Value

För \$ping krävs det bara två Headers.

Unikt värde som Postman skapar

Värdet som skriptet tog fram

# Typ av åtkomstintyg

1000 Skapa åtkomstintyg / Token apo AP Save ⌵ ⋮

**POST** ⌵ `{{webTicket}}/oauth2/Oauth2AccessTokenServlet?healthcareProfessionalLice`

Params ● Auth Headers (7) Body Pre-req. Tests ● Settings

Query Params

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	healthcareProfessionalLicense	AP	
<input checked="" type="checkbox"/>	pharmacyIdentifier	7350046611966	
<input checked="" type="checkbox"/>	healthcareProfessionalLicenseIde...	920007	
<input type="checkbox"/>	personalPrescriptionCode	9000027	
<input type="checkbox"/>	healthcareProviderId		

Informationen här kommer att ingå i åtkomstintyget.

Vissa inparametrar kan ändras men tänk på att det kan påverka vad som ska skickas med i Provenance och Practitioner.

# Sparade variabler

1000 Skapa åtkomstintyg / Token apo AP

Params • Auth Headers (7) Body Pre-req. Tests • Settings

```

1  pm.test("Status code is 200", function () {
2      pm.response.to.have.status(200);
3      // pm.globals.set("authorizationToken", responseBody.trim());
4      pm.environment.set("authorizationToken", responseBody.trim());
5      pm.environment.set('ticketFrom', 'sektorAPOTEK');
6      pm.environment.set("X-Purpose", "EXPEDIERING");
7      pm.environment.set("X-Access", "UTAN_KRAV_SAMTYCKE");
8
9      var xUserAgent = {"name":"Apotek client","info":"Apotek client to FHIR","version":"0.
      0.1","moduleVersion":"0.0.1"}
10     pm.globals.set('X-User-Agent', JSON.stringify(xUserAgent));
11     var jsonUserAgent = pm.globals.get('X-User-Agent');
12     pm.globals.set("X-User-Agent-encoded", btoa(jsonUserAgent));
13
14     var xOrgInfo = {"orgenhetsId":"7350046611966","orgenhetsIdTyp":"GLN",
      "orgenhetsOrt":"Kalmar"};
15     pm.globals.set('X-Org-Info', JSON.stringify(xOrgInfo));
16     var jsonOrgInfo = pm.globals.get('X-Org-Info');
17     pm.globals.set("X-Org-Info-encoded", btoa(jsonOrgInfo));
18 });

```

Åtkomstintyget

x-purpose  
x-access

x-user-agent

x-org-info

Efter ett lyckat anrop sätts vissa variabler som sedan används i efterföljande testfall.

Standardinställning för åtkomst och ändamål som ändras i vissa användningsfall.

# Sparade variabler

1210 Hämta patient - Alla typer av intyg / environment data byt pat

POST ▼ https://postman-echo.com/get

Params Authorization Headers (7) Body Pre-request Script Tests Settings

```

1 //växlar patient, check environment.patientPnr om pnr på den patient man vill jobba med
2 var patientPnr = null;
3 patientPnr = '<personnr utan bindestreck>';
4 pm.environment.set("patientPnr", patientPnr);
5 console.log('patientPnr=', pm.environment.get("patientPnr"), 'patientId', pm.environment.get
  ("patientId"));
6
7 //1 kör PatientWithPnr för att s
8 //2 kör patientWithId kontroll o
9

```

Vilken patient man vill använda kan sättas via användningsfallet "environment data byt pat" eller direkt på sidan som visar alla miljövariabler.



Testprojekt Ext18

	VARIABLE	TYPE ⓘ	INITIAL VALUE ⓘ
<input checked="" type="checkbox"/>	patientPnr	default <span>▼</span>	

# Byta åtkomst och ändamål

1210 Hämta patient - Alla typer av intyg / patientWithPnr

GET  `{{base}}/Patient?identifier={{pnrSystemUri}}{{patientPnr}}`

Params ● Authorization ● Headers (11) Body Pre-request Script ● Tests ●

```
1  if(pm.environment.get('X-Purpose') == 'VARD') {
2    pm.environment.set('X-Purpose', 'VARD');
3    pm.environment.set('X-Access', 'UTAN_KRAV_SAMTYCKE');
4  } else if(pm.environment.get('X-Purpose') == 'EXPEDIERING') {
5    pm.environment.set('X-Purpose', 'EXPEDIERING');
6    pm.environment.set('X-Access', 'UTAN_KRAV_SAMTYCKE');
7  } else if(pm.environment.get('X-Purpose') == 'LASA_EGNA_UPPGIFTER') {
8    pm.environment.set('X-Purpose', 'LASA_EGNA_UPPGIFTER');
9    pm.environment.set('X-Access', 'EGNA_UPPGIFTER');
10 }
11
```

Användningsfallet kräver  
andra inställningar för  
åtkomst och ändamål

# Återställa åtkomst och ändamål

```
1210 Hämta patient - Alla typer av intyg / patientWithPnr

GET {{base}}/Patient?identifier={{pnrSystemUri}}{{patientPnr}}

Params • Authorization • Headers (11) Body Pre-request Script • Tests

1
2 //set tillbaka som i VARD TICKET
3 if(pm.environment.get('tickeFrom') == 'sektorVARD' || pm.environment.get('tickeFrom') == 'VARD') {
4     pm.environment.set('X-Purpose', 'VARD');
5     pm.environment.set('X-Access', 'TILLFALLIGT_SAMTYCKE');
6 }
7
```

Återställning av åtkomst och ändamål till de inställningar som sattes när åtkomstintyget skapas.

# Logiskt ID för patienten

1210 Hämta patient - Alla typer av intyg / patientWithPnr

GET ▼ `{{base}}/Patient?identifier={{pnrSystemUri}}{{patientPnr}}`

Params ● Authorization ● Headers (11) Body Pre-request Script ● Tests ●

```
7
8 var statusOk = Boolean(pm.response.to.have.status(200));
9 pm.test("Status code is 200", function () {
10   pm.response.to.have.status(200);
11   var jsonData = JSON.parse(responseBody);
12   var resource = jsonData.entry[0].resource;
13   if(resource != null && resource.resourceType == 'Patient') {
14     var patientPnr = resource.identifier[0].value;
15     pm.variables.get("variable_key");
16     var patRef = resource.id;
17     console.log('patientPnr=', patientPnr, 'patientId=', patRef);
18     pm.environment.set("patientId", patRef);
19     pm.environment.set("patientPnr", patientPnr);
20   }
21 });
22
```

Det är samma programkod som i patientWithID vilket förklarar varför även patientId sparas.

Patientresursens logiska ID sparas om anropet gick bra och sedan används det av efterföljande användningsfall.

# Skapa och ändra

- ▼ 1330 Skapa åtkomstsamtycke - Vård
  - POST Token vard LK
  - > GET get consent by UUID
  - POST jsonPractitioner LK
  - POST jsonProvenance vard
  - POST vard create accessConSent

Kopia på motsvarande AF i 1000 – Åtkomstintyg

Kopia på motsvarande AF i 1320 – Hämta åtkomstsamtycke

Sparar Practitioner för efterföljande AF

Sparar Provenance för efterföljande AF

Skapar en begäran om åtkomstsamtycke

# Person som gör anropet

1330 Skapa åtkomstsamtal - Vård / jsonPractitioner LK

POST <https://postman-echo.com/post> Send

Params Auth Headers (7) Body Pre-req. Tests Settings

```
83  "qualification": [  
84    {  
85      "identifier": [  
86        {  
87          "system": "urn:oid:1.2.752.116.3.1.2",  
88          "value": "9000027"  
89        }  
90      ],  
91      "code": {  
92        "coding": [  
93          {  
94            "system": "urn:oid:1.2.752.116.3.1.3",  
95            "code": "LK"  
96          }  
97        ]  
98      }  
99    }  
100  ]
```

Samma förskrivningskod som i åtkomstintyget

Samma yrkeskod som i åtkomstintyget

# Person som gör anropet



1330 Skapa ätkomstsamtal - Vård / jsonPractitioner LK

POST <https://postman-echo.com/post> Send

Params Auth Headers (7) Body Pre-req Tests Settings

```
105 pm.environment.set('jsonPractitioner', JSON.stringify(jsonPractitioner));
106 pm.environment.set('jsonPractitionerVard', JSON.stringify(jsonPractitioner));
107 //console.log(pm.environment.get('jsonPractitioner'));
108 var jp = JSON.parse(pm.environment.get('jsonPractitioner'));
109 //console.log(jp);
```

```
104     "qualification": {
105       {
106         "identifier": {
107           {
108             "system": "urn:oid:1.2.752.116.3.1.3",
109             "code": "LK"
110           }
111         }
112       }
113     }
114   }
115 }
```

Practitioner-resursen  
sparas för  
efterföljande AF



# Sammanfattning

- Globala variabler sätts via skript.
- Miljövariabler sätts också via skript förutom följande.
  - Den testmiljö man vill använda.
  - Det läkemedel/handelsvara man vill använda.
  - Personnr man vill använda om det inte sätts via ett skript.
- Det går att skapa skript som körs före och efter ett anrop.
  - Skript som ska köras för varje användningsfall kan läggas på Collection-nivå. Det finns ett sådant skript i 1150 – Anslutningstest.
  - Det finns skript i de flesta användningsfallen som sparar och återanvänder data såsom Logiskt ID för patienten, åtkomst, ändamål, Practitioner och Provenance mm.

# Agenda

- Inledning
- Genomgång av strukturen i Postman
- **Praktiska övningar**
  - **Förberedelser**
  - Köra några av användningsfallen
  - Skapa nya användningsfall
  - Egna labbar
- Avslutning

# Förberedelser

- Lägg in ett av era tilldelade personnr i Postman.
- Kör anropen i Collection 1210 – Hämta patient.
  - Kolla vilka variabler som har sparats globalt och i Ext18-miljön.
  - Prova att använda åtkomstintyg för vård, apotek och privatperson.

# Förskrivare per organisation

Organisation	Förskrivarkod 1	Förskrivarkod 2
Alfa e-care	9001140	9001173
Apotea	9001140	9001173
Appva	9001181	9001199
Cambio	9001181	9001199
Cerner	9001207	9001215
CGM	9001207	9001215
Extenda retail	9001249	9001256
Inera	9001249	9001256
Kliniken Patientsystem	9001363	9002098
Kry	9001363	9002098
LS retail	9002106	8456113
Muntra	9002106	8456113
Receptum	9001025	9001033
TietoEvry	9001025	9001033

Saknas någon organisation?



# Agenda

- Inledning
- Genomgång av strukturen i Postman
- **Praktiska övningar**
  - Förberedelser
  - **Köra några av användningsfallen**
  - **Skapa nya användningsfall**
  - Egna labbar
- Avslutning

# Köra några AF

Tillämpa befintliga användningsfall.

- 1330 – Skapa åtkomstsamtycke
  - Börja med att kolla så det inte finns några samtycken.
  - Kolla igenom de AF som finns i 1330 och se om något skript behöver ändras.
- 1320 – Hämta åtkomstsamtycke
  - När samtycket är skapat kör 1320 "search access consent ? patRef & category & practitioner.id" och titta på resultatet.
  - Hämta samtycket både som privatperson och som samtyckestagare. Prova gärna olika kombinationer av sökparametrar.

Tips!

Skapa en kopia på en hel Collection innan du börjar ändra eller kopiera enskilda AF.

Tips!

I Simplifier finns det information över giltiga kombinationer av sökparametrar.

# Köra några AF

Tillämpa befintliga användningsfall fortsättning.

- Testa även att göra följande
  - Försök att skapa ett till samtycke för samma samtyckestagare (förskrivarkod 9000027).
  - Skapa ett samtycke för en annan samtyckestagare (förskrivarkod 9001207) och hämta alla samtycken som privatperson och båda samtyckestagarna. Notera skillnaderna.

# Skapa nya AF

Skapa nya användningsfall.

- 1720 – Skapa förskrivning
  - Skapa ett nytt AF baserat på ”create medReq tillfällesdosering”.
  - Byt doseringsinstruktion till ett som motsvarar exempel 11 ”Frequence administration with intervals” i Simplifier.
  - Kolla igenom de AF som finns i 1720 och se om något skript behöver ändras.
- 1710 – Hämta förskrivning
  - Skapa en kopia på ett AF i hämta förskrivning och hämta alla förskrivningar som är aktiva och utfärdade idag.

Den skapade förskrivningen kommer att användas i nästa övning.

# Skapa nya AF

Ändra skripten.

- 1720 – Skapa förskrivning
  - Innan vi kör AF 1460 behöver förskrivningskedjeid sparas.
  - Kolla skripten i AF 1710 "read medReq uuid" och kopiera den del som sparar förskrivningskedjeid.
  - Lägg in programkoden och skapa en ny förskrivning.
  - Kolla att det förskrivningskedjeid som finns i den nya förskrivningen är det som har sparats i miljövariabeln `createdNLLPrescriptionChain`.

Den skapade förskrivningen kommer att användas i nästa övning.

# Köra några till AF

Tillämpa befintliga användningsfall.

- 1460 – Skapa sekretesspärr
  - Kolla igenom de AF som finns i 1460 och se om något skript behöver ändras.
  - Kör sedan “vard registred datalock behandlingsorsak SEKRETESS\_PAT”.
- 1440 – Hämta spärr
  - När spärren är skapad hämta den som vård, apotek och privatperson. Notera skillnaderna.

# Köra några till AF

Tillämpa befintliga användningsfall.

- 1464 – Skapa integritetsspärr
  - Kolla igenom de AF som finns i 1464 och se om något skript behöver ändras.
  - Kör sedan “vard-pat registred datalock behandlingsorsak INTEGRITET”.
- 1440 – Hämta spärr
  - När spärren är skapad hämta den som vård, apotek och privatperson. Notera skillnaderna.

# Agenda

- Inledning
- Genomgång av strukturen i Postman
- **Praktiska övningar**
  - Förberedelser
  - Köra några av användningsfallen
  - Skapa nya användningsfall
- **Egna labbar**
- Avslutning

# Egna labbar

Om tid finns fortsätter vi med egna laborationer.

# Agenda

- Inledning
- Genomgång av strukturen i Postman
- **Praktiska övningar**
  - Förberedelser
  - Köra några av användningsfallen
  - Skapa nya användningsfall
  - Egna labbar
- **Avslutning**

# Framöver

- Ingen plan hos EHM att vidareutveckla Postmanprojektet
- Förslagsvis ett uppföljningsmöte om några veckor för att följa upp erfarenheter och uppslag

# Reflektioner på dagens Connectathon

1. Vad har varit bra?
2. Vad kan förbättras?
3. Tankar framåt?



# Connect!

**Material:**

[Connectathon - 2023-02-09](#)





**Tack**