1. Common Error Code For Livn Api Calls

request

Method: any
URI: any

Livn api server

response

request body

404
parameter
invalid

400
request
invalid

500
internal
error

403
forbidden

timestamp
message
details
code
stacktrace

timestamp
message
details
code
stacktrace

timestamp
details
code
stacktrace

timestamp
status(403)
error
path

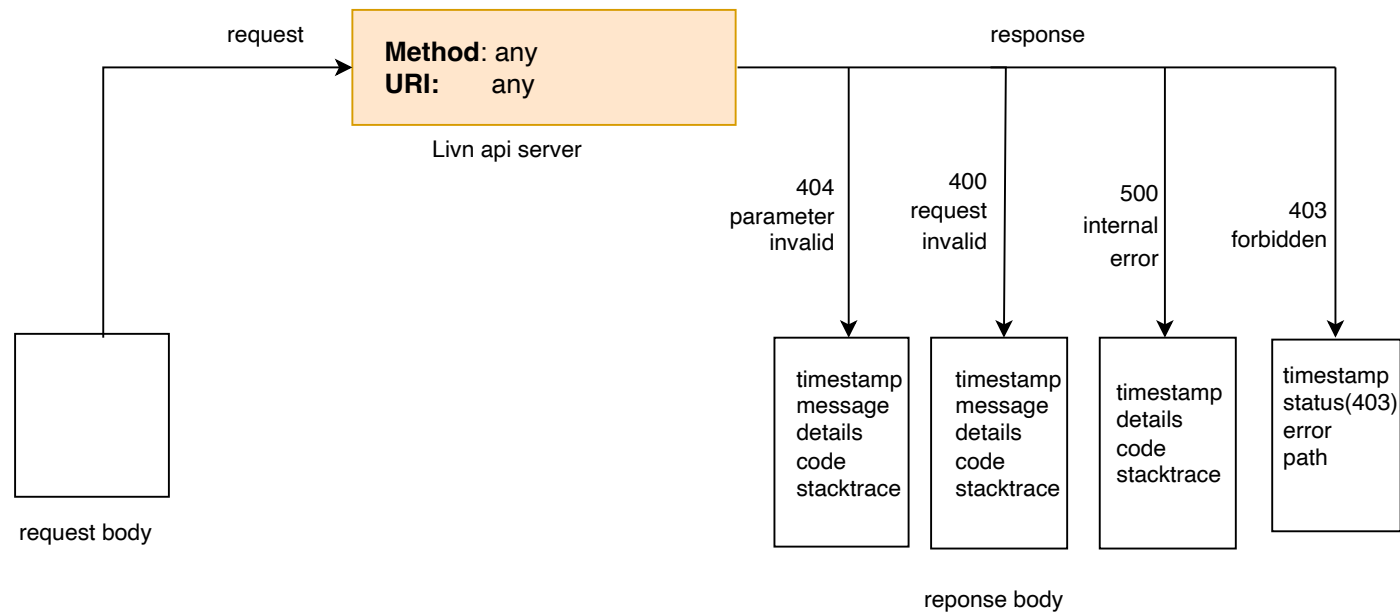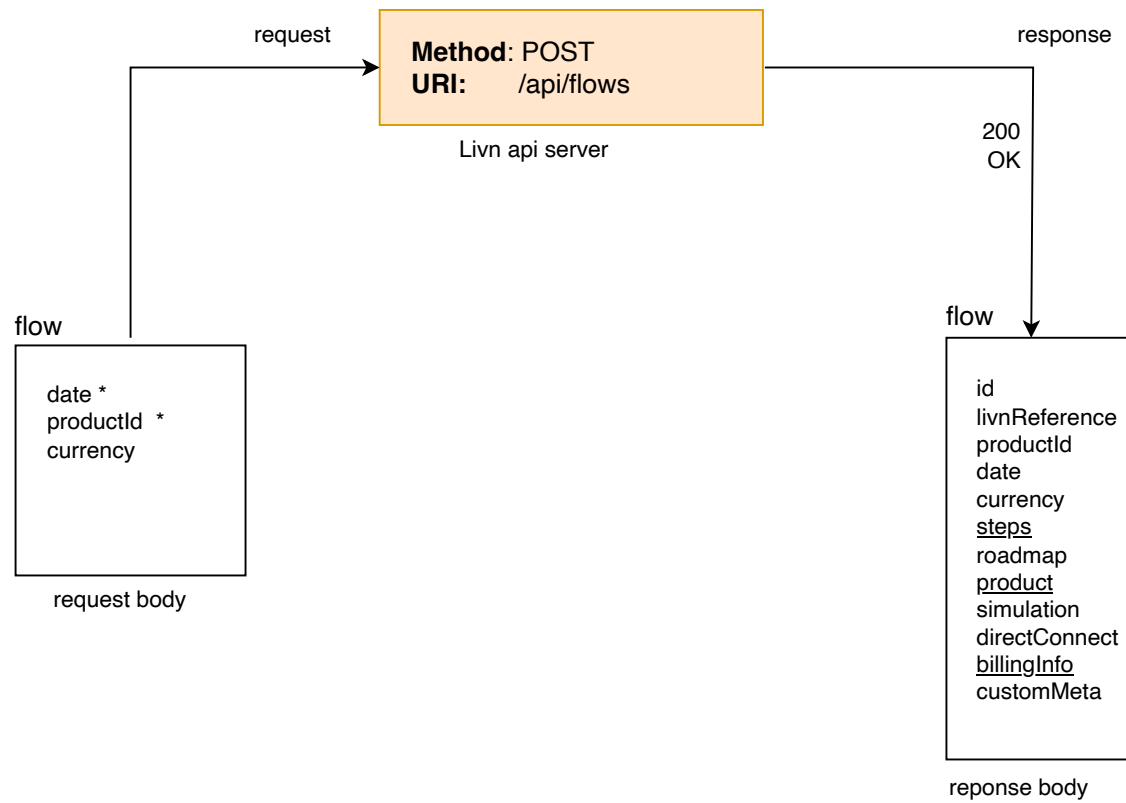reponse body

Note

1. These errors are common and can happen to any Livn api call. They will not be listed in the following api call diagram again.

2. These errors are developer-level error, no need to display to user. If UI and front-end code works well, these error would not normally happen. If they do happen it means bugs in the front-end code or server code. In this case, display a message saying "System error, please try again later." or something like this. There will be log record this call in the server program which can be analyzed later.

3. For errors that caused by user input (user-level error), they will be covered later in this document.

4. Some example reason for causing these errors.
    404 e.g. initiate a flow without product id or with invalid date.
    400 e.g. make a request with json data that can not be parsed.
    403 e.g. make api call without authorization data.

2. Initiate a Flow

request   **Method**: POST   response

URI: /api/flows

Livn api server

200
OK

flow

```
date *
productId  *
currency
```

request body

flow

```
id
livnReference
productId
date
currency
steps
roadmap
product
simulation
directConnect
billingInfo
customMeta
```
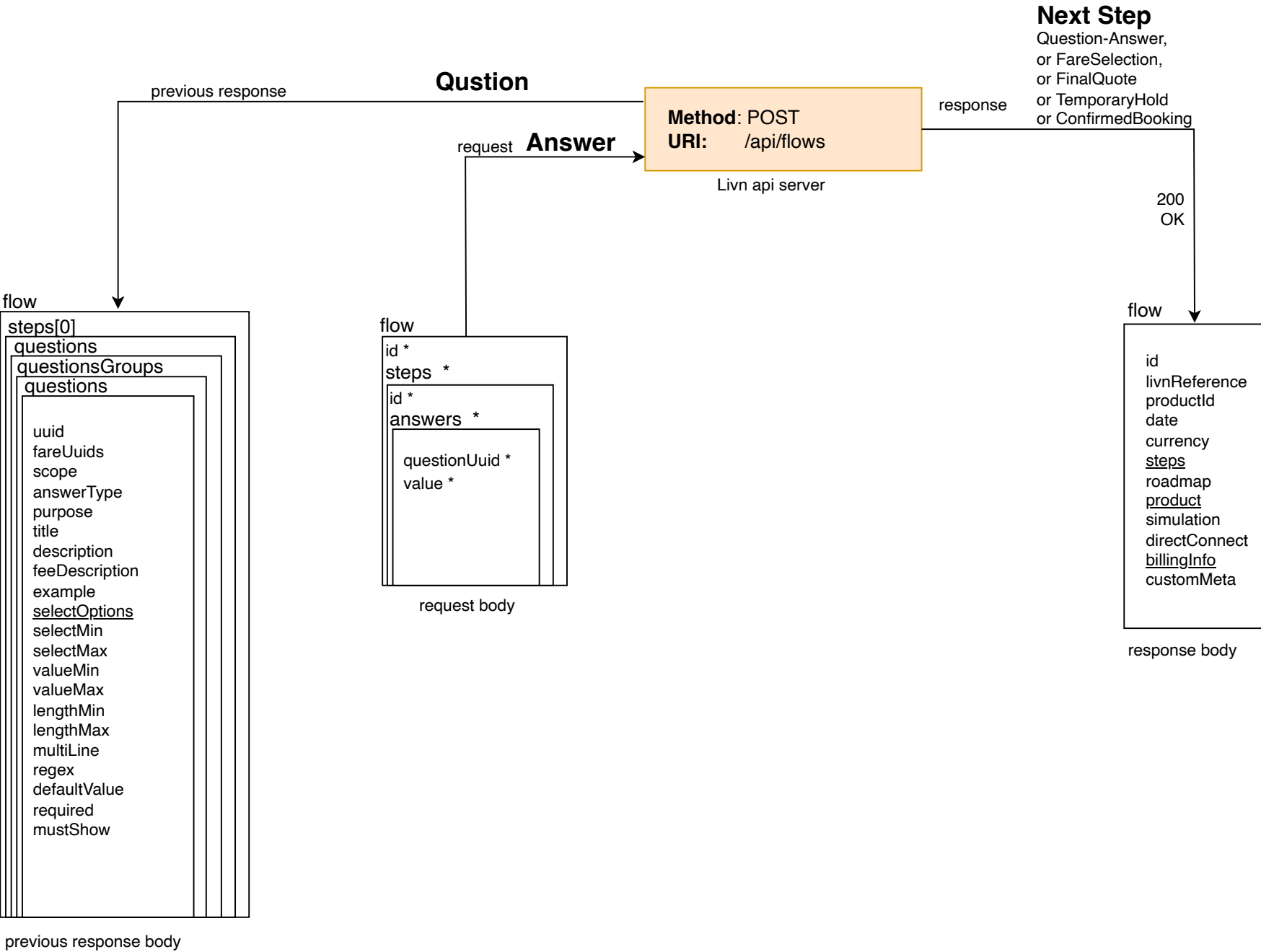
reponse body

Note

1. This is the 1st step to book a product

2. After initiate a flow successfully, check the following response data

```
if (flow.steps[0].stepName == "FareSelection") {
        //go on with the FareSelection step
}else{
        //go on with the Question-Answer step
}
```

3. Question-Answer

Next Step
Question-Answer,
or FareSelection,
or FinalQuote
or TemporaryHold
or ConfirmedBooking

previous response

Qustion

request **Answer**

response

Method: POST
URI: /api/flows

Livn api server

200
OK

flow

steps[0]

questions

questionsGroups

questions

uuid
fareUuids
scope
answerType
purpose
title
description
feeDescription
example
selectOptions
selectMin
selectMax
valueMin
valueMax
lengthMin
lengthMax
multiLine
regex
defaultValue
required
mustShow

previous response body

flow

id *

steps *

id *

answers *

questionUuid *
value *

request body

flow

id
livnReference
productId
date
currency
steps
roadmap
product
simulation
directConnect
billingInfo
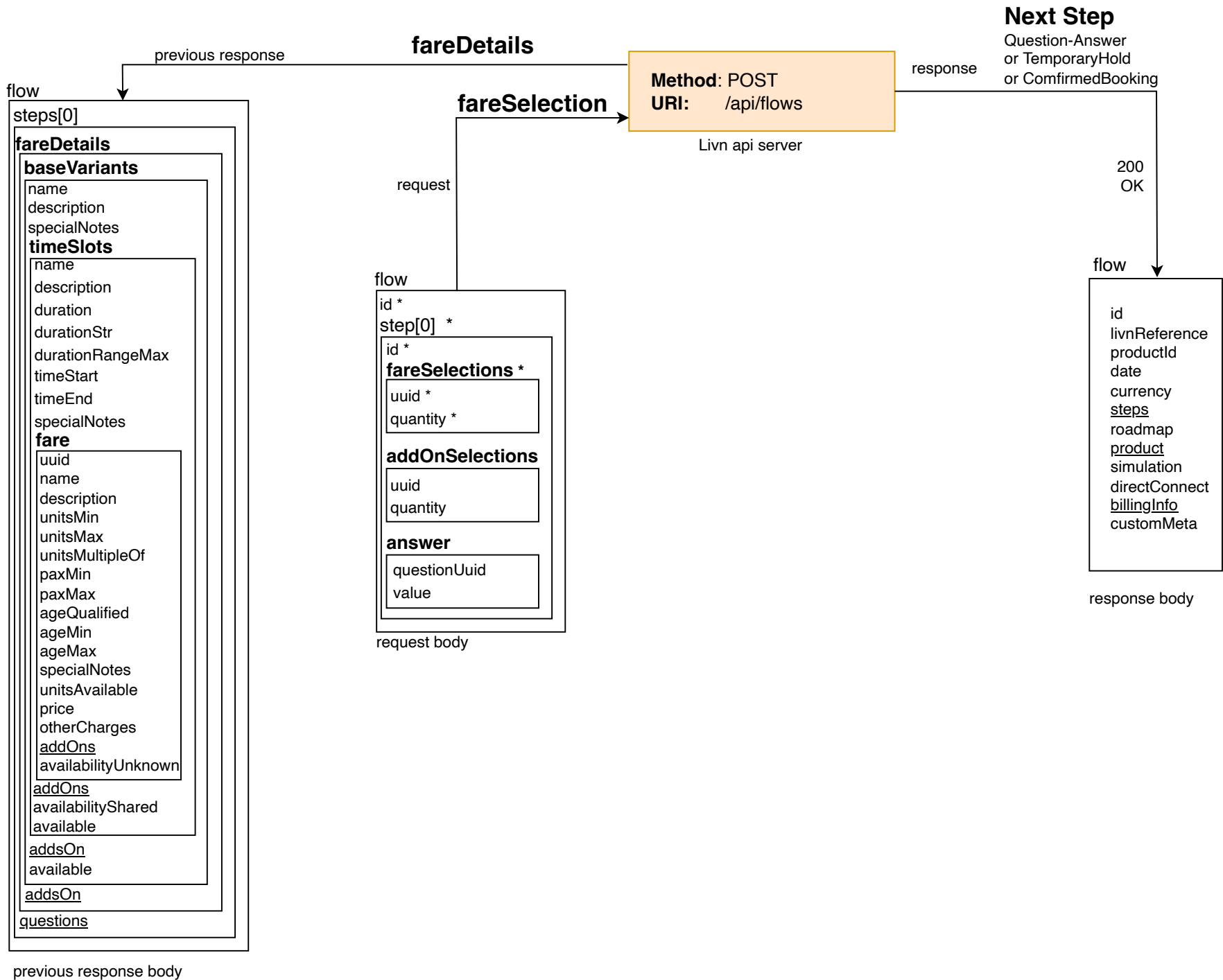customMeta

response body

Note

1. The previous response include all the questions need to be answered. The UI will be displayed by this.

2. For each question with `required==true,` must be answered with valid values by sending a request to the server.

3. Before sending the answers, all values need to be validated using the validating properties if provided.

4. After the request has been sent, if get response, the response should be carefully checked.
check the value of status in step[0].

```
if(steps[0].status == "FAILED"){
  //questions are not properly answered/user input is invalid
  if(!steps[0].error.terminateTransaction && !goBackToRetry){
    //ask user to re-answer the questions
  }else if(!step[0].error.terminateTransaction && goBackToRetry){
    //go back and retry
  }else{
    //abandon this flow
  }
}else if(steps[0].status == "ACTIVE"){
  //a new step is entered, check step[0].stepName to see whether we are entering to Q-A, FareSelection, FinalQuote or TEMPORARY_HOLD or
  //ConfirmedBooking step
}else if(steps[0].status == "DONE"){
  //current booking is finished.
}
```

5. The FinalQuote milestone can be seen as a normal Question-Answer step with finalQuote data. Both the questions and and finalQuote should be displayed to the user.

6. Check the following link for question display
https://github.com/RobinOnTheWay/livn-doc/blob/main/UI%20Rendering%20and%20User%20Input%20Validation%20Using%20Livn%20API%20Data.md#1-display-questions

7. Check the following link for answer validation.
https://github.com/RobinOnTheWay/livn-doc/blob/main/UI%20Rendering%20and%20User%20Input%20Validation%20Using%20Livn%20API%20Data.md#1-display-questions

4. FareSelection

previous response ------ **fareDetails** ------→

Next Step
Question-Answer
or TemporaryHold
or ComfirmedBooking

fareSelection ------→

Method: POST
URI: /api/flows

Livn api server

response

200
OK

flow
steps[0]

fareDetails

baseVariants

name
description
specialNotes

timeSlots

name
description
duration
durationStr
durationRangeMax
timeStart
timeEnd
specialNotes

fare

uuid
name
description
unitsMin
unitsMax
unitsMultipleOf
paxMin
paxMax
ageQualified
ageMin
ageMax
specialNotes
unitsAvailable
price
otherCharges
addOns
availabilityUnknown

addOns
availabilityShared
available

addsOn
available

addsOn

questions

previous response body

request

flow

id *
step[0] *

id *

fareSelections *

uuid *
quantity *

addOnSelections

uuid
quantity

answer

questionUuid
value

request body

flow

id
livnReference
productId
date
currency
steps
roadmap
product
simulation
directConnect
billingInfo
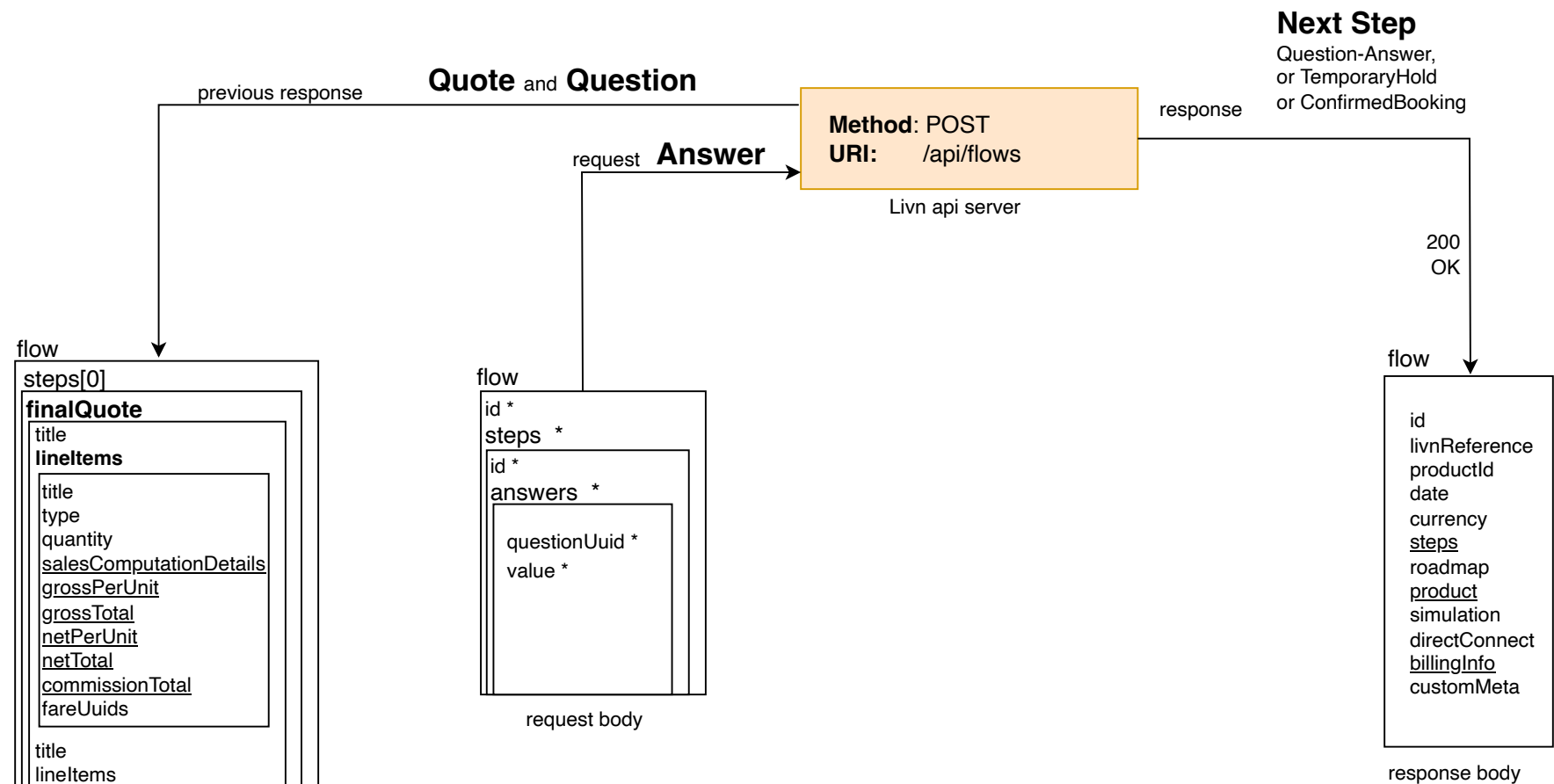customMeta

response body

Note

1. The previous response include all fareDetails, optional addOns and questions need to be selected or answered. The UI will be displayed by this.

2. Only available baseVariants, timeSlots, fares, addOns need to be displayed. (`available == true` or `unitsAvailable > 0`)

3. Before sending the fareSelection, addOnSelection, answers, all values need to be validated using the validating properties if provided.

4. After the request has been sent, if get response, the response should be carefully checked.
check the value of status in step[0].

```
if(steps[0].status == "FAILED"){
  //fareSelections/addOnSelections/answers are not properly provided/user input is invalid
  if(!steps[0].error.terminateTransaction && !goBackToRetry){
    //ask user to redo the fareSelection
  }else if(!step[0].error.terminateTransaction && goBackToRetry){
    //go back and retry
  }else{
    //abandon this flow
  }
}else if(steps[0].status == "ACTIVE"){
  //a new step is entered, check step[0].stepName to see whether we are entering to Q-A, FinalQuote or TEMPORARY_HOLD or
 //ConfirmedBooking step
}else if(steps[0].status == "DONE"){
  //current booking is finished.
}
```

Once the user made the FareSelection, the price can be previewed anytime by calling the previewPrice API
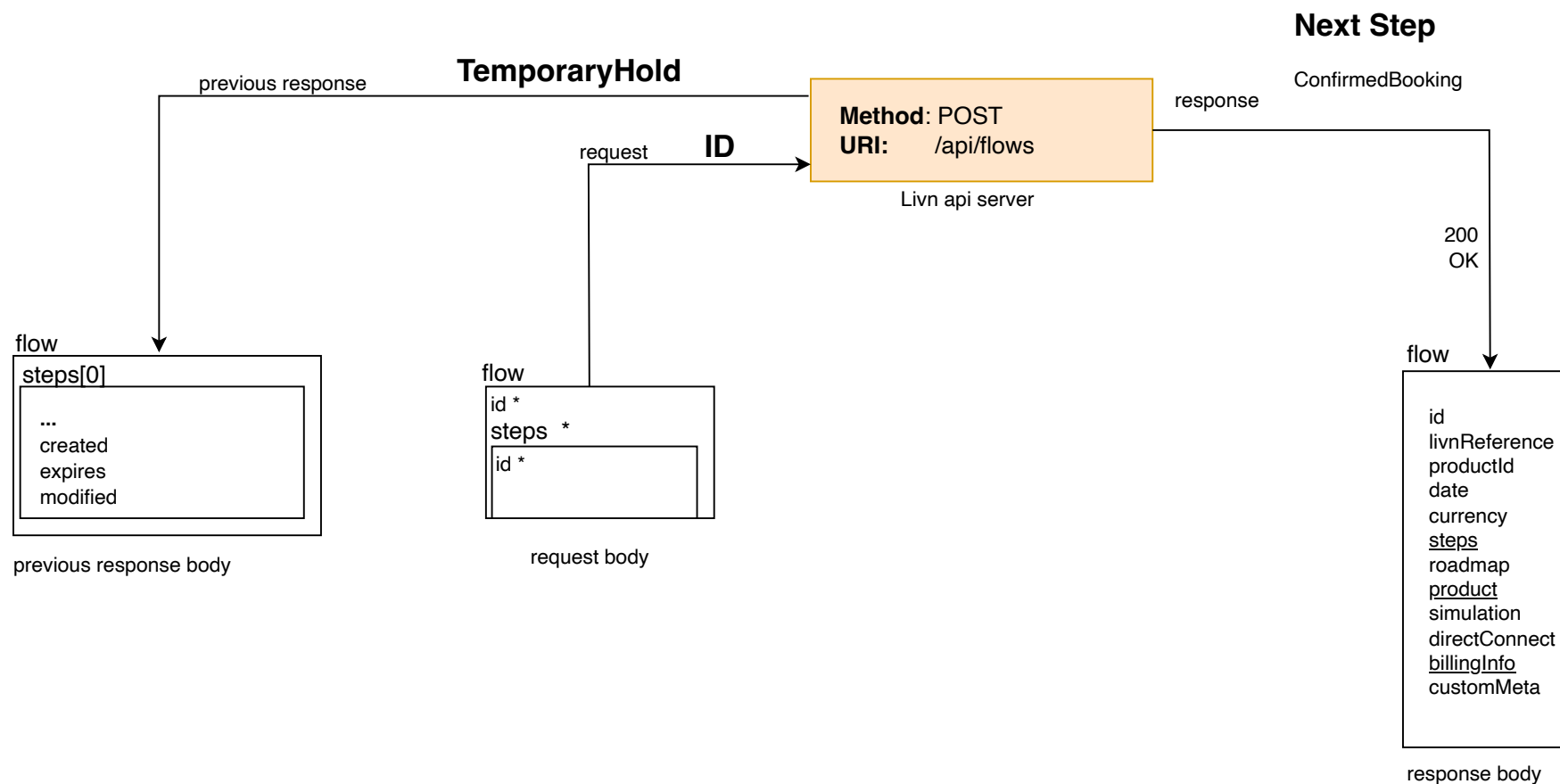
5. FinalQuote

Next Step
Question-Answer,
or TemporaryHold
or ConfirmedBooking

Quote and **Question**

previous response

request **Answer**

response

```
Method: POST
URI:     /api/flows
```
Livn api server

200
OK

flow

steps[0]

finalQuote

title
lineItems

title
type
quantity
salesComputationDetails
grossPerUnit
grossTotal
netPerUnit
netTotal
commissionTotal
fareUuids

title
lineItems
cancellationPolicy
localFees
grossTotal
netTotal
contractCommTotal
resSuppliedCommTotal
generalTerms

questions

...

created
expires
modified

previous response body

flow

id *
steps *

id *
answers *

questionUuid *
value *

request body

flow

id
livnReference
productId
date
currency
steps
roadmap
product
simulation
directConnect
billingInfo
customMeta

response body

Note

1. The finalQuote milestone can be seen as normal Q-A step with `finalQuote` data which must be displayed to user.

2. The questions and answers should be displayed and validated as in the Q-A step.

3. Be careful that the FinalQuote step usually come with `expires` time. Once the final quote expires we usually need to go back (usually to the FareSelection step) and proceed the flow. All the other steps can also come with expires, always check whether this property is exist in the current step.
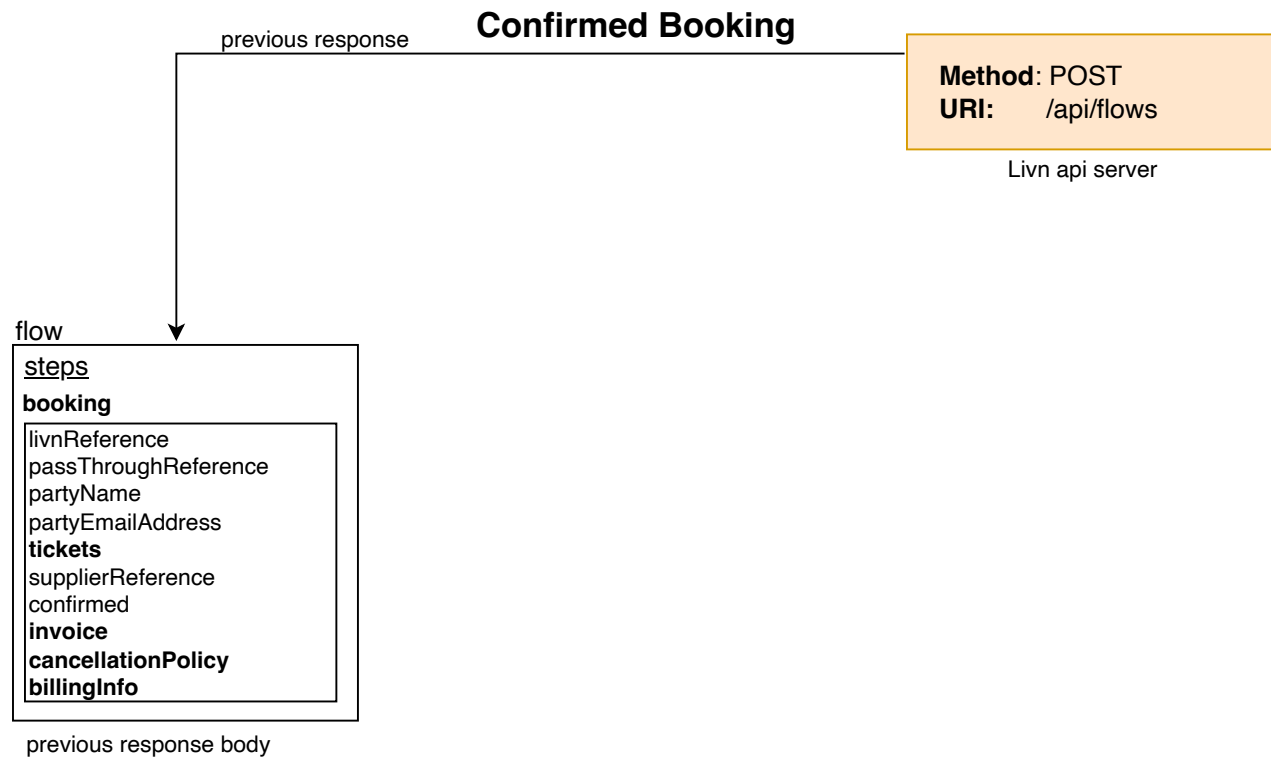
6. Temporary Hold

Next Step

previous response

TemporaryHold

ConfirmedBooking

response

request **ID**

Method: POST
URI: /api/flows

Livn api server

200
OK

flow

steps[0]

...
created
expires
modified

flow

id *
steps *
id *

flow

id
livnReference
productId
date
currency
steps
roadmap
product
simulation
directConnect
billingInfo
customMeta

previous response body

request body

response body

Note

1. The optional `TemporaryHold` milestone come with no `fareDetails` or `Questions`, so only flow id and step id (no `fareSelections` or `answers`) need to be sent back.

3. Be careful that the temporaryHold step usually also come with `expires` time. Once this step expires we need to go back to proceed the flow.
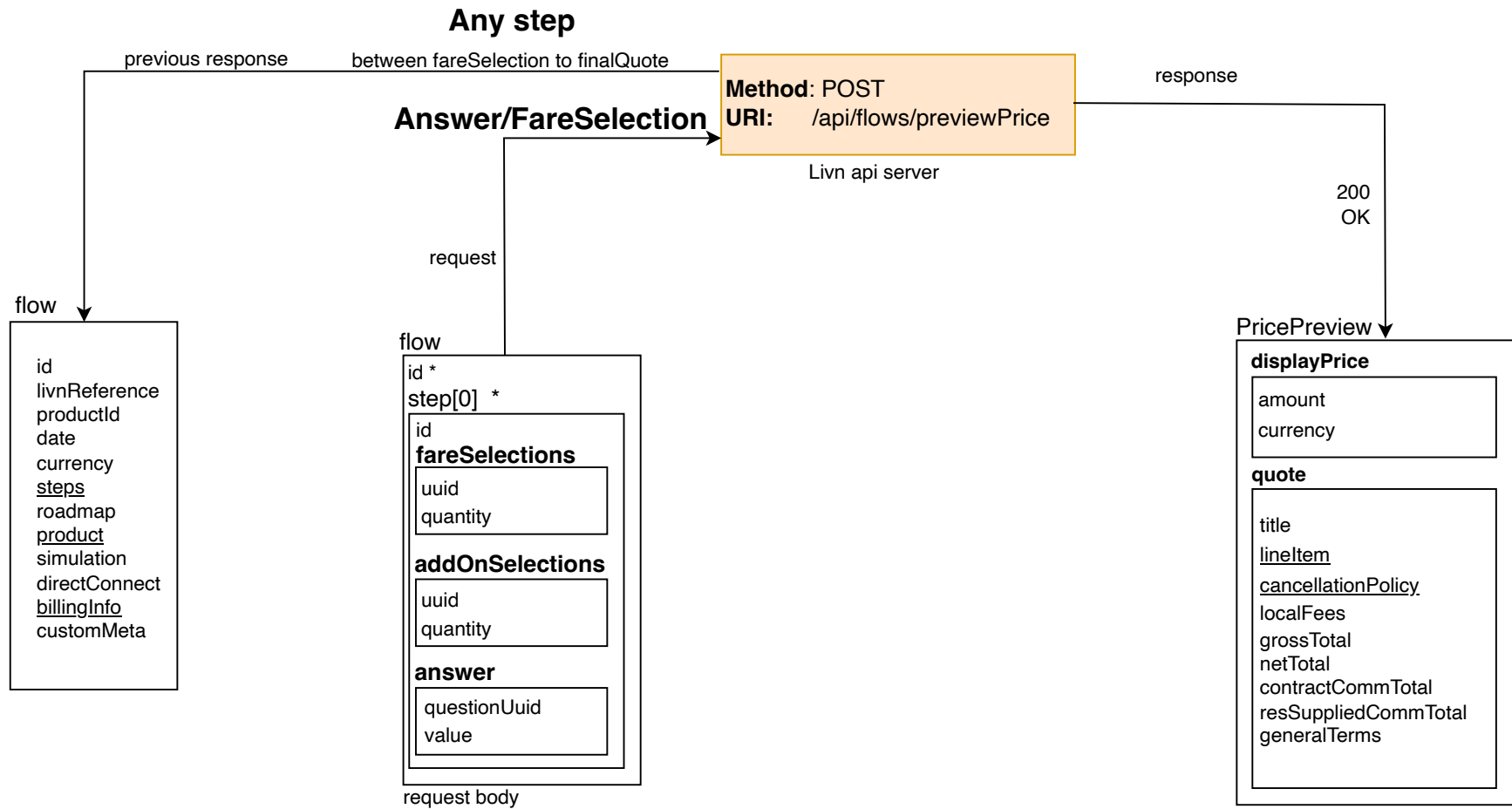
6. Confirmed Booking

Confirmed Booking

previous response

Confirmed Booking
Method: POST
URI: /api/flows

Livn api server

flow

steps
booking
livnReference passThroughReference partyName partyEmailAddress **tickets** supplierReference confirmed **invoice** **cancellationPolicy** **billingInfo**

previous response body

Note

1. When a previous step is finished with a request send back to LivnApi server, it is possible to enter this Confirmed Booking step. When this step is reached, it means that the whole flow has come to an end successfully.

2. This step can be entered from a FareSelection, Question-Answer, or TemporaryHold step.

3. When this step is reached, there will be `booking` data included in the response. The booking id can be then used in the booking related APIs.
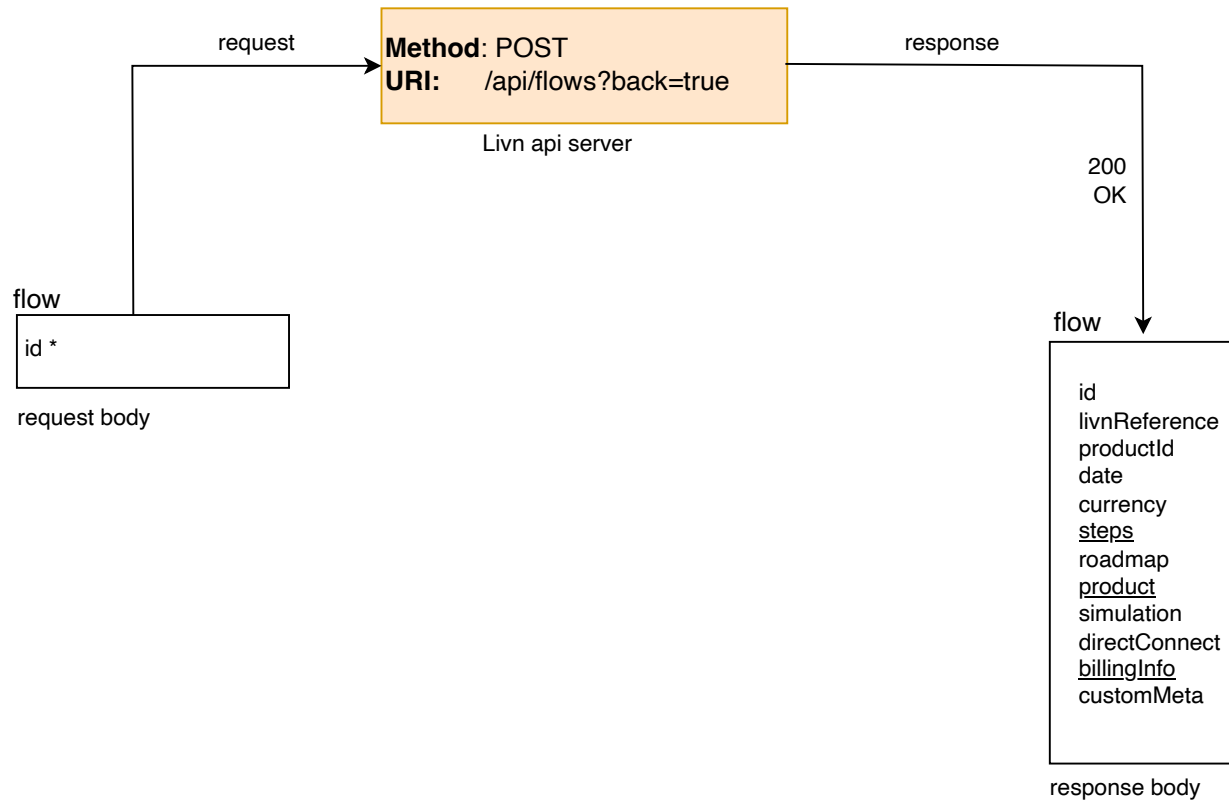
7. Price Preview

Any step

previous response   between fareSelection to finalQuote   response

Answer/FareSelection | **Method**: POST
                          | **URI:**   /api/flows/previewPrice

Livn api server

200
OK

request

flow

| id |
| livnReference |
| productId |
| date |
| currency |
| steps |
| roadmap |
| product |
| simulation |
| directConnect |
| billingInfo |
| customMeta |

flow

id *

step[0] *

id

fareSelections

| uuid |
| quantity |

addOnSelections

| uuid |
| quantity |

answer

| questionUuid |
| value |

request body

PricePreview

displayPrice

| amount |
| currency |

quote

| title |
| lineItem |
| cancellationPolicy |
| localFees |
| grossTotal |
| netTotal |
| contractCommTotal |
| resSuppliedCommTotal |
| generalTerms |

Note

1. This API can be called between fareSelection to finalQuote steps.

2. The request body is the same you would use for current step to POST to /api/flows.
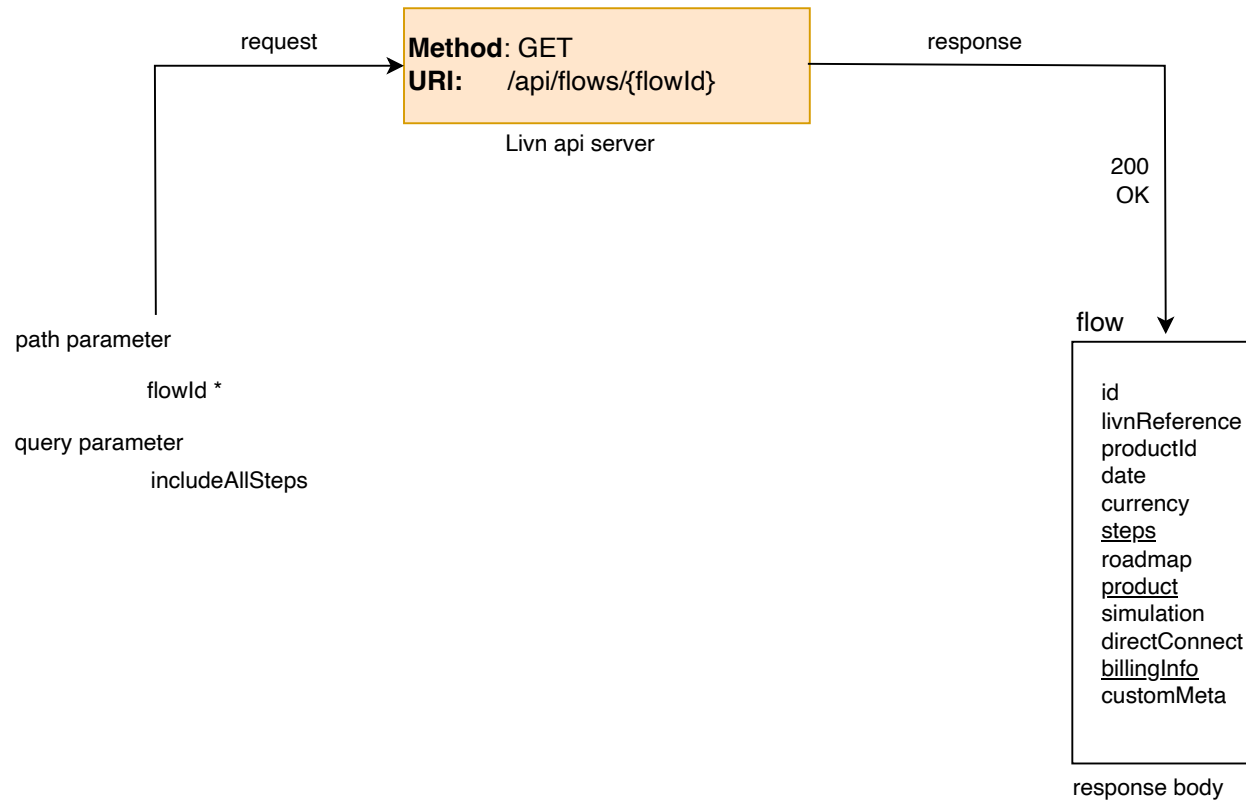
8. Go Back


```
Method: POST
URI:    /api/flows?back=true
```
Livn api server

request

flow

id *

request body

response

200
OK

flow

id livnReference productId date currency steps roadmap product simulation directConnect billingInfo customMeta

response body

Note

1. Possible reasons for call this API could be user's attention/expired step/error handling with value of goBackToRetry is set as true.

2. Only steps with `allowBackHere == true` can be moved backwards to, the steps between will be skipped.

3. When a flow is just initiated or a flow has been confirmed, it is not possible to move back to a previous step.

9. Retrieve Flow

request	**Method**: GET **URI:** /api/flows/{flowId}	response

Livn api server

200
OK

path parameter

flowId *

query parameter

includeAllSteps

flow

id livnReference productId date currency steps roadmap product simulation directConnect billingInfo customMeta

response body

Note

1. Once the flow is initiated, it remain in existence even if the booking is confirmed or the confirmed booking is cancelled.