Visual driven database queries

What if you want to do powerful stuff and you are using HTML

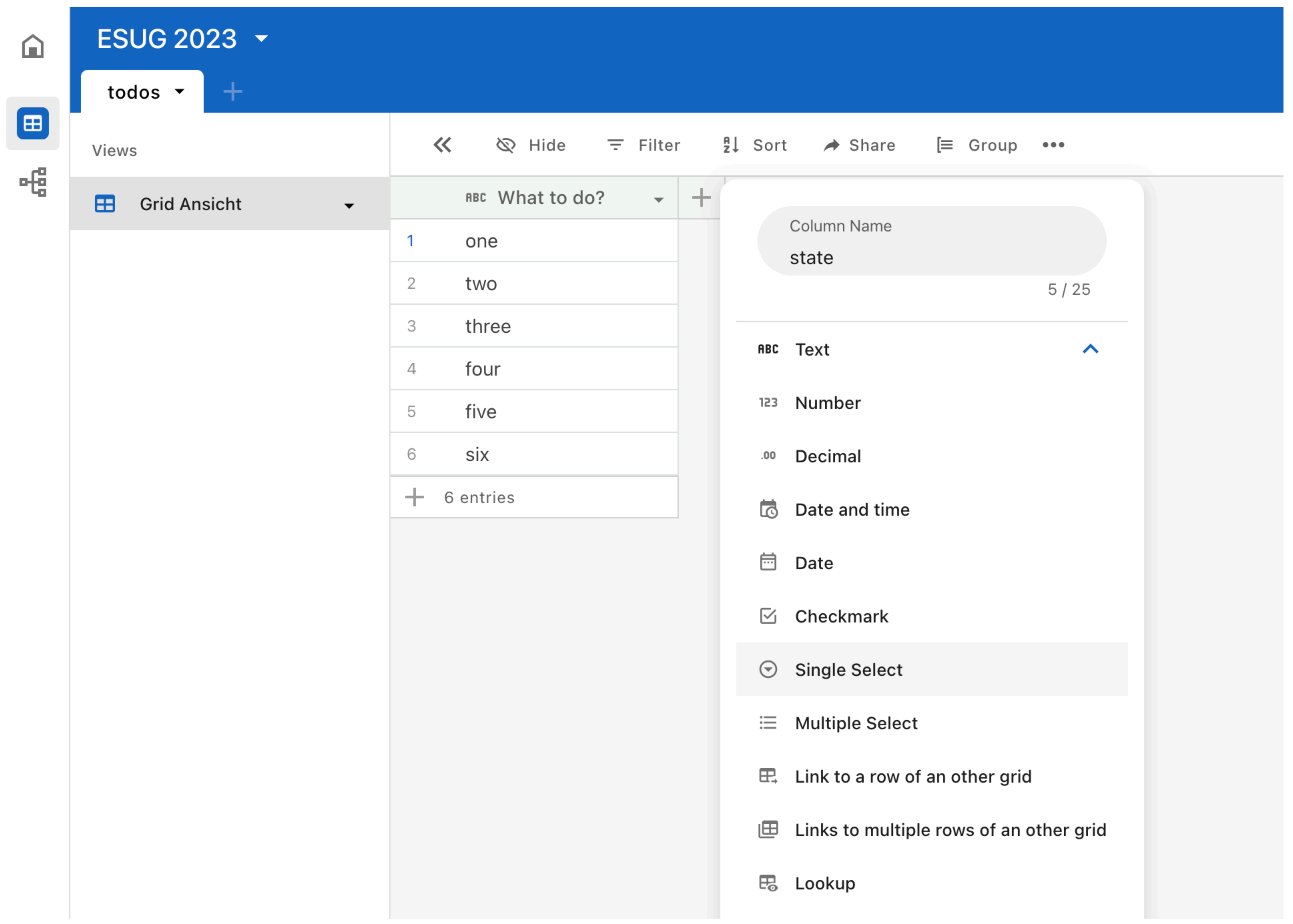
Norbert Hartl, ApptiveGrid GmbH, ESUG 2023

ApptiveGrid

A visual database and flow management tool

visual database

Defining a model



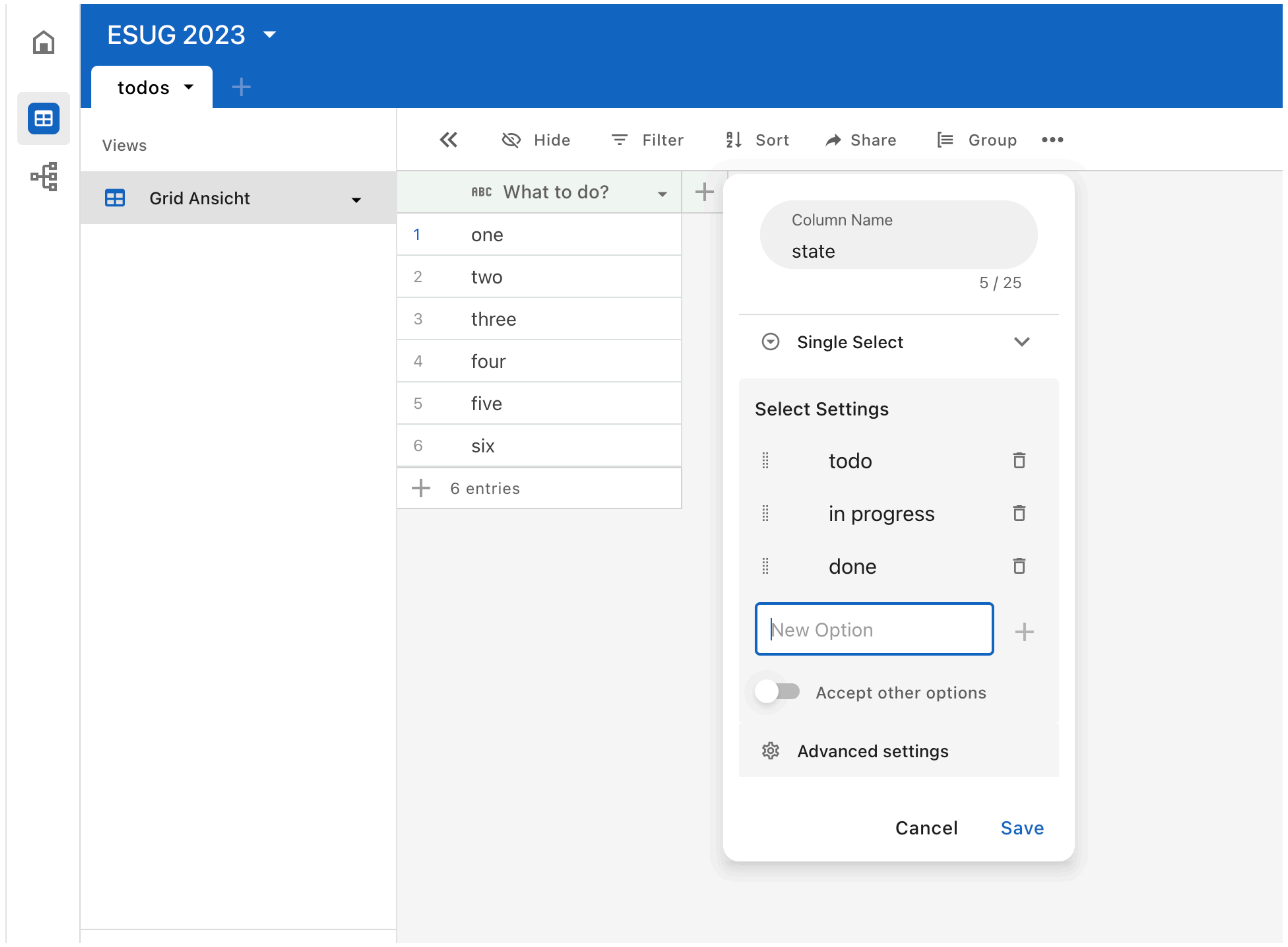
	What to do?
1	one
2	two
3	three
4	four
5	five
6	six

6 entries

Column Name
state
5 / 25

Text
Number
Decimal
Date and time
Date
Checkmark
Single Select
Multiple Select
Link to a row of an other grid
Links to multiple rows of an other grid
Lookup

Defining a model - adding an enum


ESUG 2023

todos

Views

Grid Ansicht

Hide Filter Sort Share Group

	What to do?
1	one
2	two
3	three
4	four
5	five
6	six

6 entries

Column Name
state
5 / 25

Single Select

Select Settings

todo
in progress
done

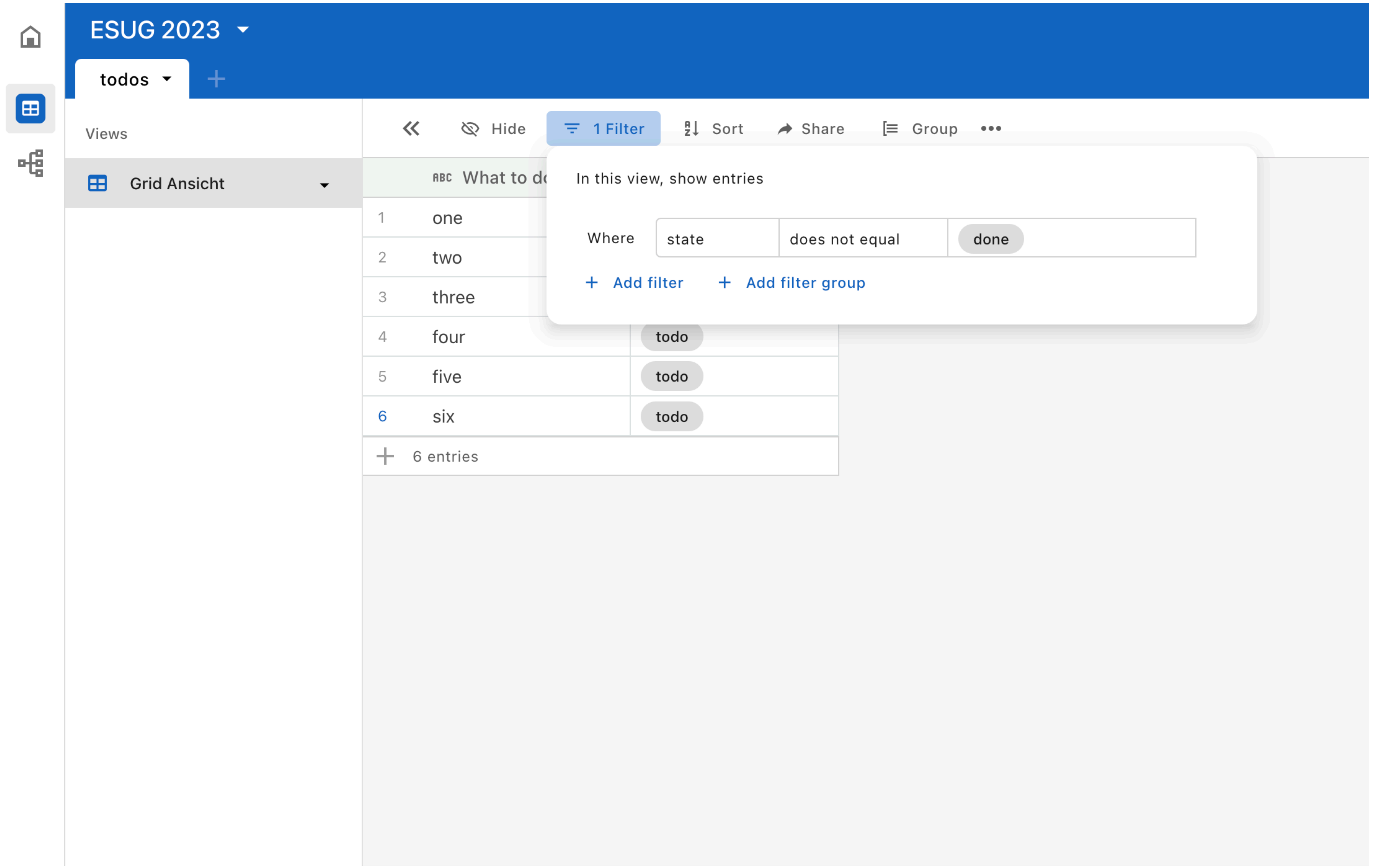
New Option

Accept other options

Advanced settings

Cancel Save

Filtering & sorting



ESUG 2023

todos

Views

Grid Ansicht

Hide | 1 Filter | Sort | Share | Group

In this view, show entries

Where | state | does not equal | done

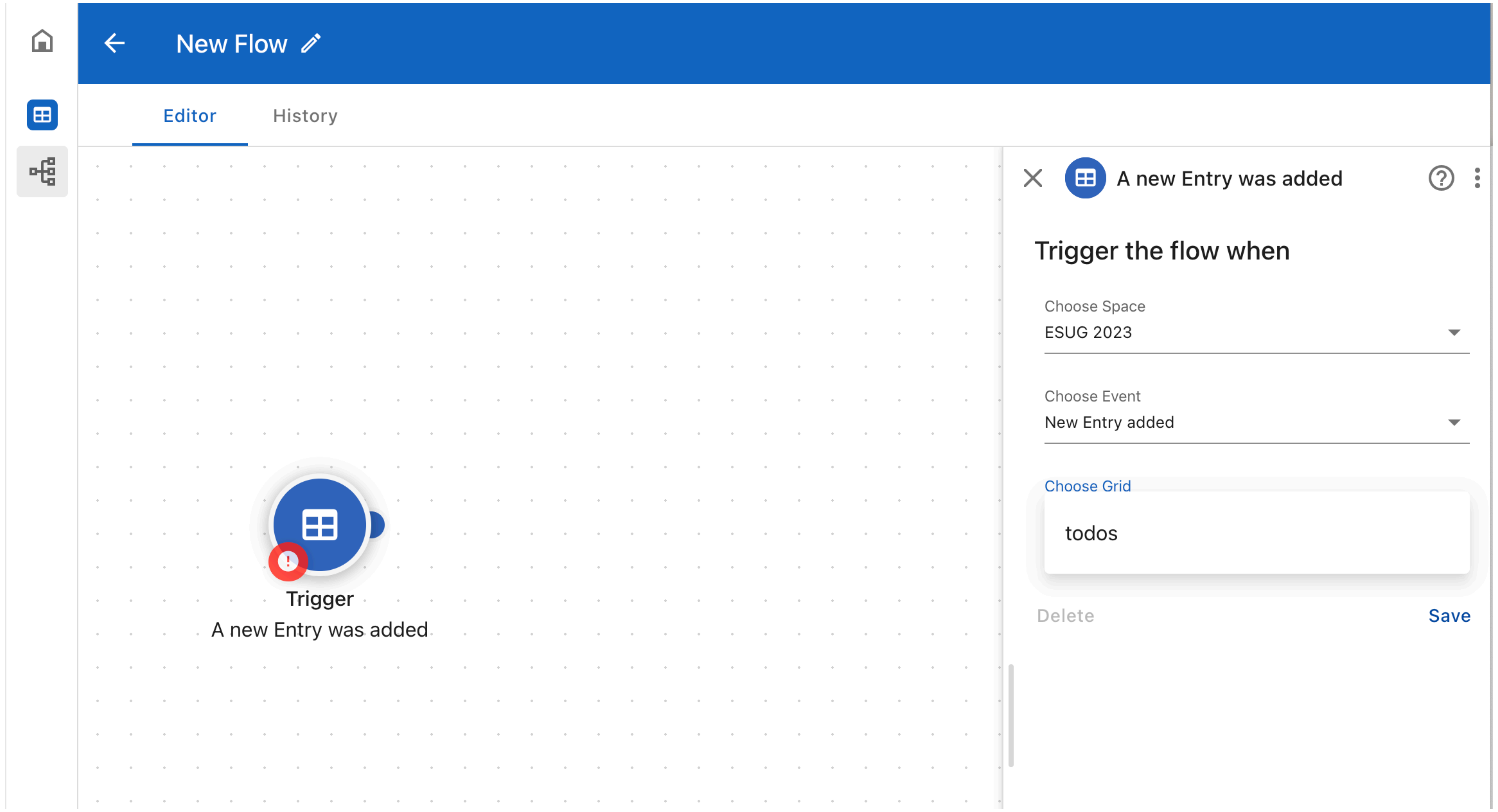
+ Add filter + Add filter group

	What to d…	
1	one	
2	two	
3	three	
4	four	todo
5	five	todo
6	six	todo

6 entries

workflow engine

Defining a flow - a trigger



A new Entry was added

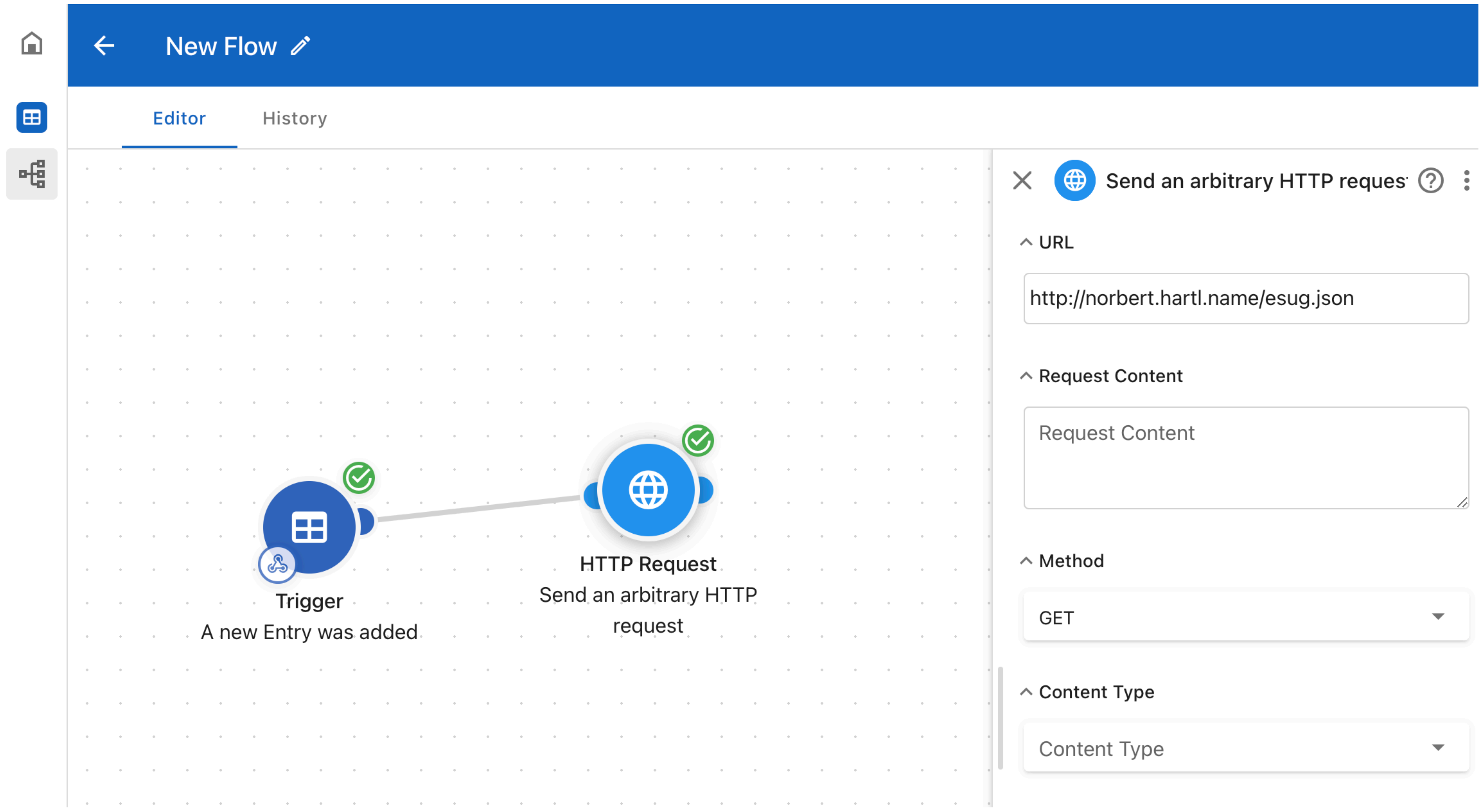
Trigger the flow when

Choose Space: ESUG 2023

Choose Event: New Entry added

Choose Grid: todos

Defining a flow - do an external call



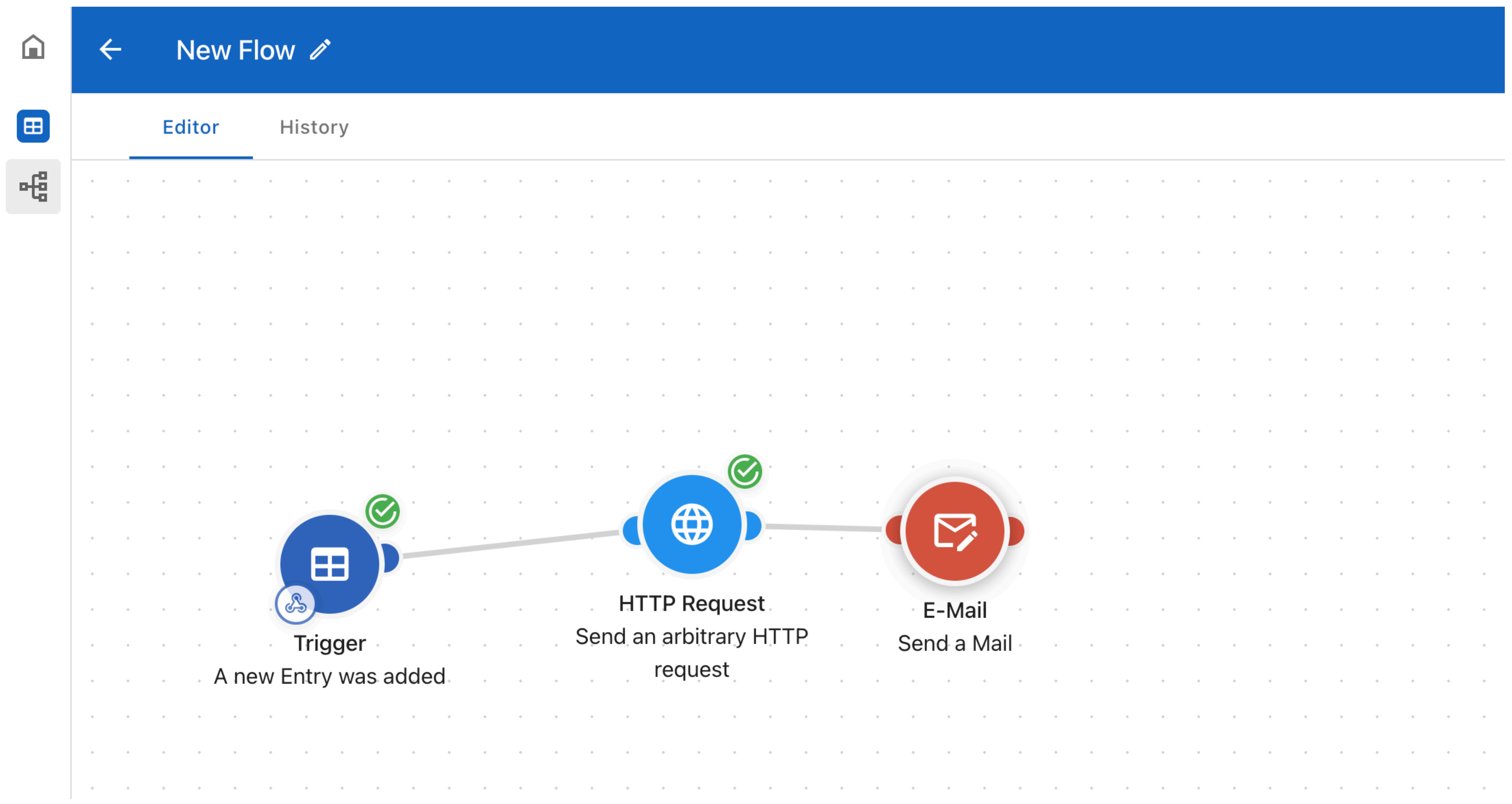
URL: http://norbert.hartl.name/esug.json

Request Content: Request Content

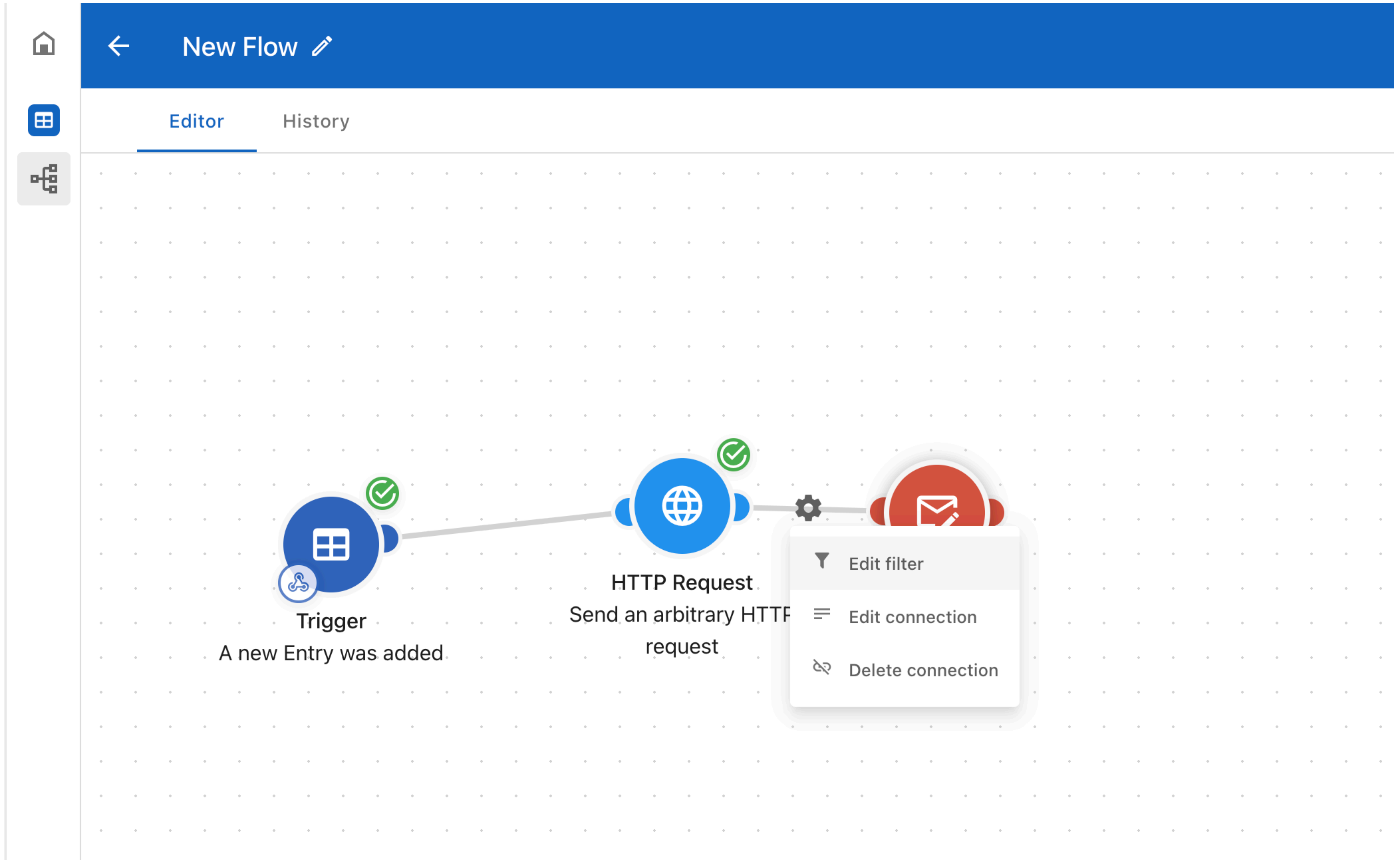
Method: GET

Content Type: Content Type

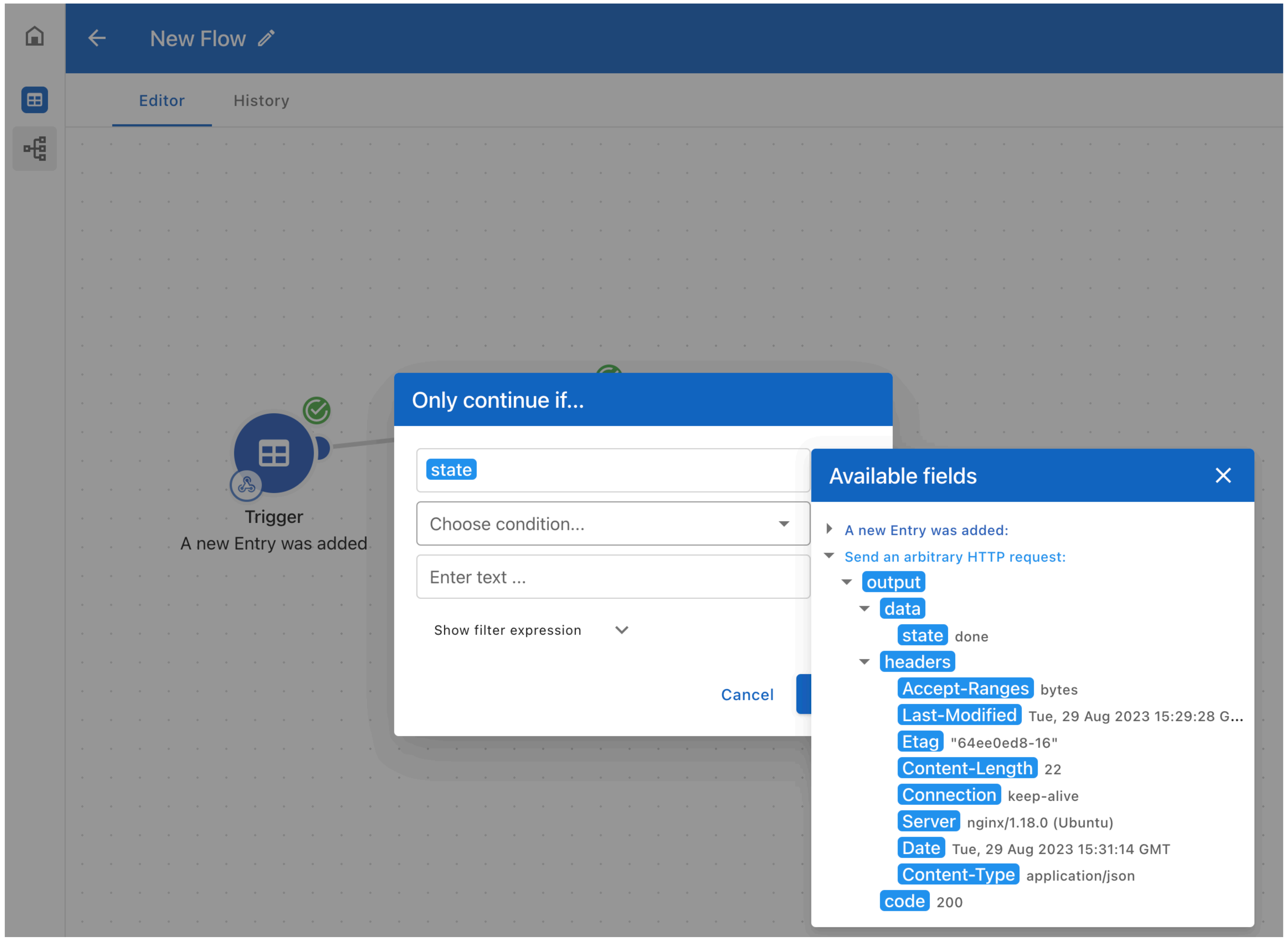
Defining a flow - send a mail ...



Defining a flow - ... with condition...

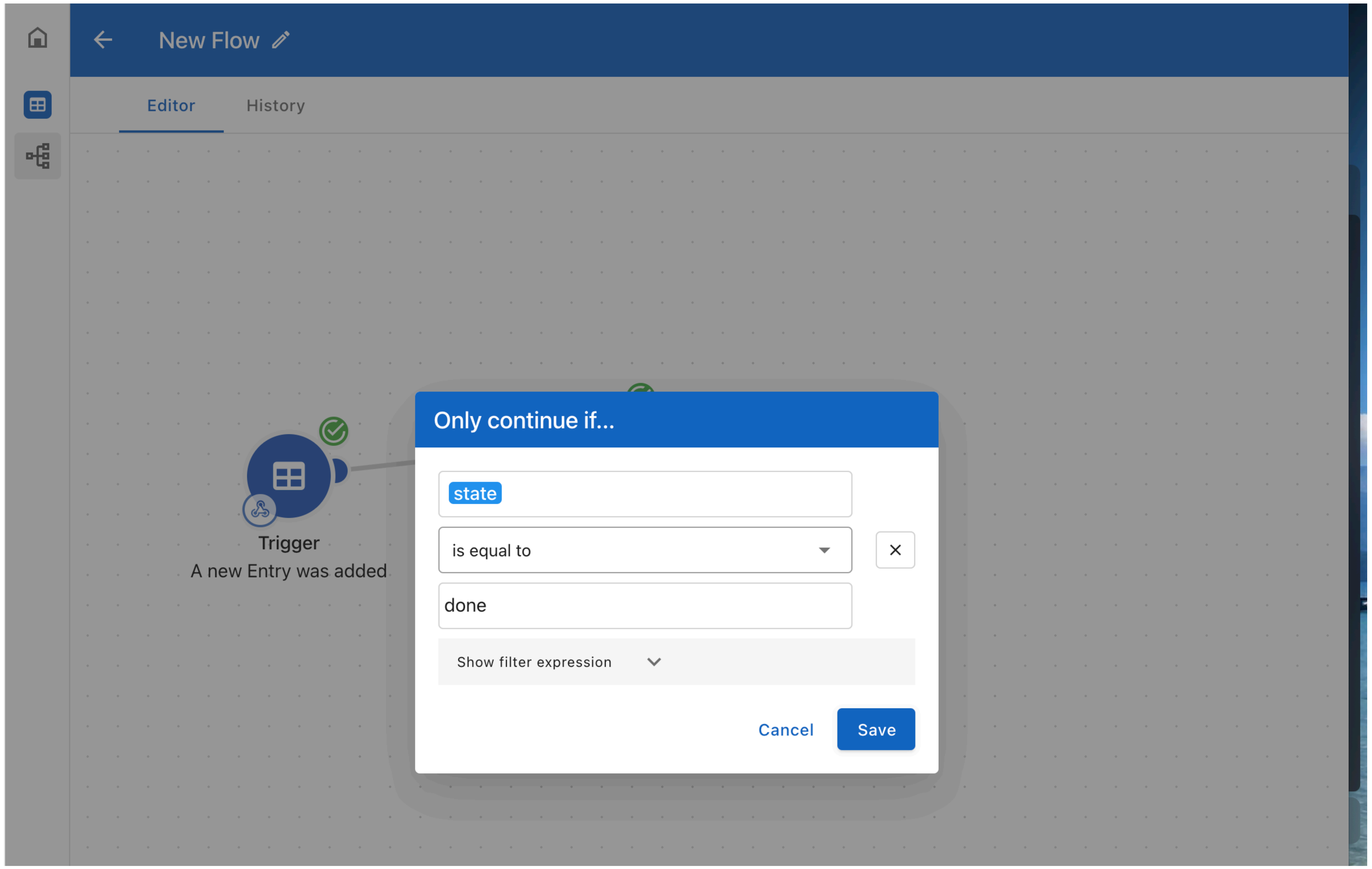


Defining a flow - visually pick whatever you got



A new Entry was added:
Send an arbitrary HTTP request:
output
data
state done
headers
Accept-Ranges bytes
Last-Modified Tue, 29 Aug 2023 15:29:28 G...
Etag "64ee0ed8-16"
Content-Length 22
Connection keep-alive
Server nginx/1.18.0 (Ubuntu)
Date Tue, 29 Aug 2023 15:31:14 GMT
Content-Type application/json
code 200

Defining a flow - define condition



state
is equal to
done

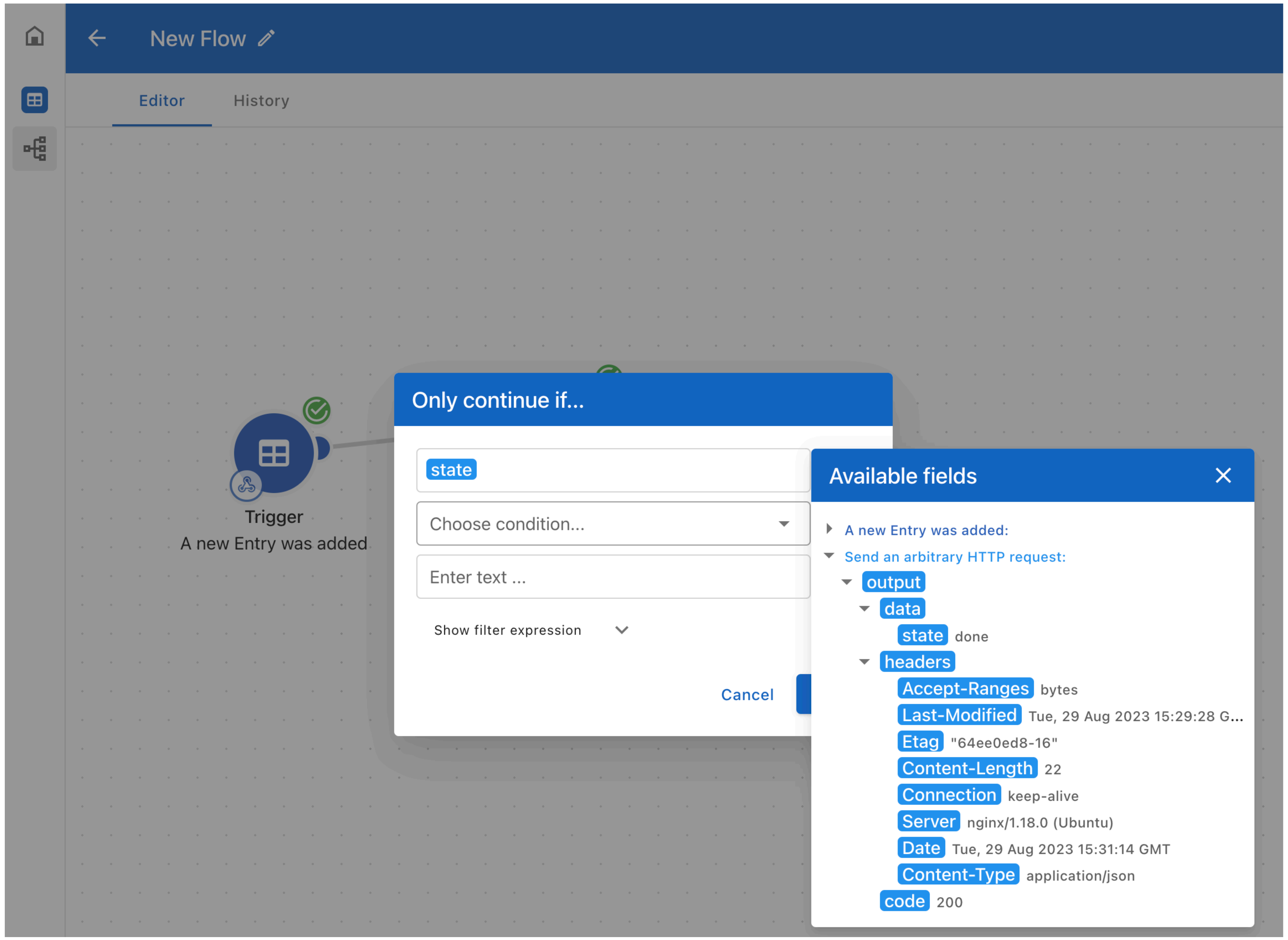
Uniform approach

- has knowledge about column types and operations
- has syntax for composing type based filters
- turns JSON into a tree that generates access path
- knows about application scope

ApptiveScript

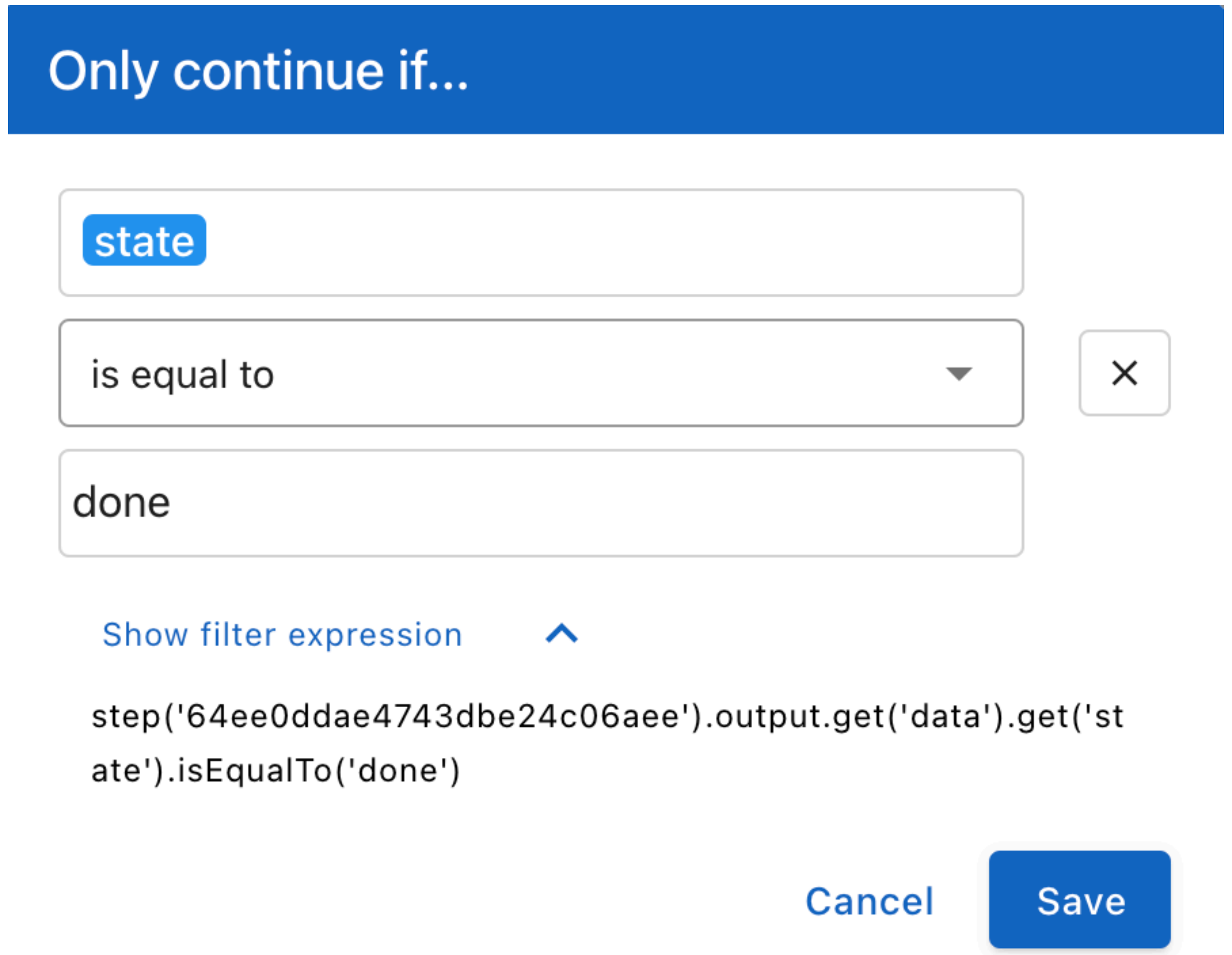
- has knowledge about column types and operations
- has syntax for composing type based filters
- turns JSON into a tree that generates access path
- knows about application scope
- is an executable DSL for ApptiveGrid

Defining a flow - visually pick whatever you got



A new Entry was added:
Send an arbitrary HTTP request:
output
data
state done
headers
Accept-Ranges bytes
Last-Modified Tue, 29 Aug 2023 15:29:28 G...
Etag "64ee0ed8-16"
Content-Length 22
Connection keep-alive
Server nginx/1.18.0 (Ubuntu)
Date Tue, 29 Aug 2023 15:31:14 GMT
Content-Type application/json
code 200

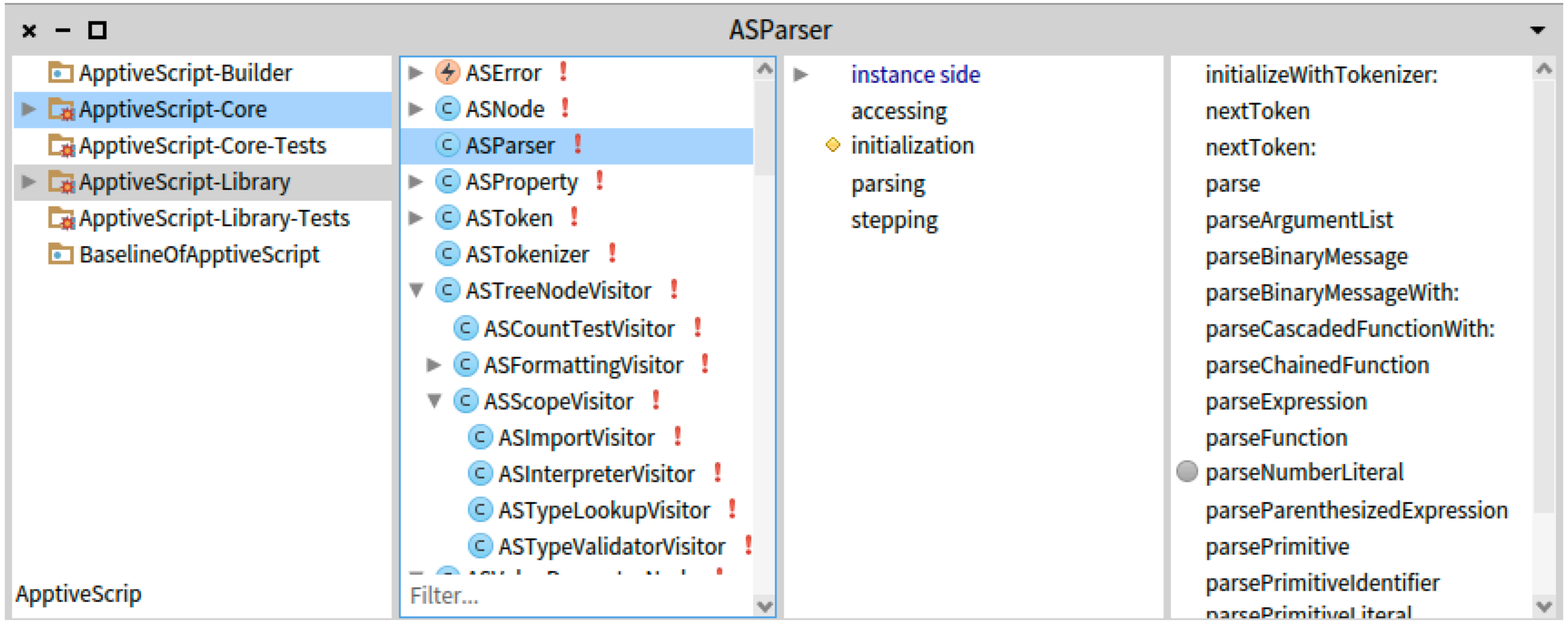
Combined syntax



```
step('64ee0ddae4743dbe24c06aee').output.get('data').get('st
ate').isEqualTo('done')
```

$$\underbrace{\texttt{step('64ee0ddae4743dbe24c06aee')}}_{\text{application}}\underbrace{\texttt{.output.get('data')}}_{\text{node}}\underbrace{\texttt{.get('state')}}_{\text{user}}\underbrace{\texttt{.isEqualTo('done')}}_{\text{type}}$$

ApptiveScript



ApptiveScript-Builder
ApptiveScript-Core
ApptiveScript-Core-Tests
ApptiveScript-Library
ApptiveScript-Library-Tests
BaselineOfApptiveScript

ApptiveScrip

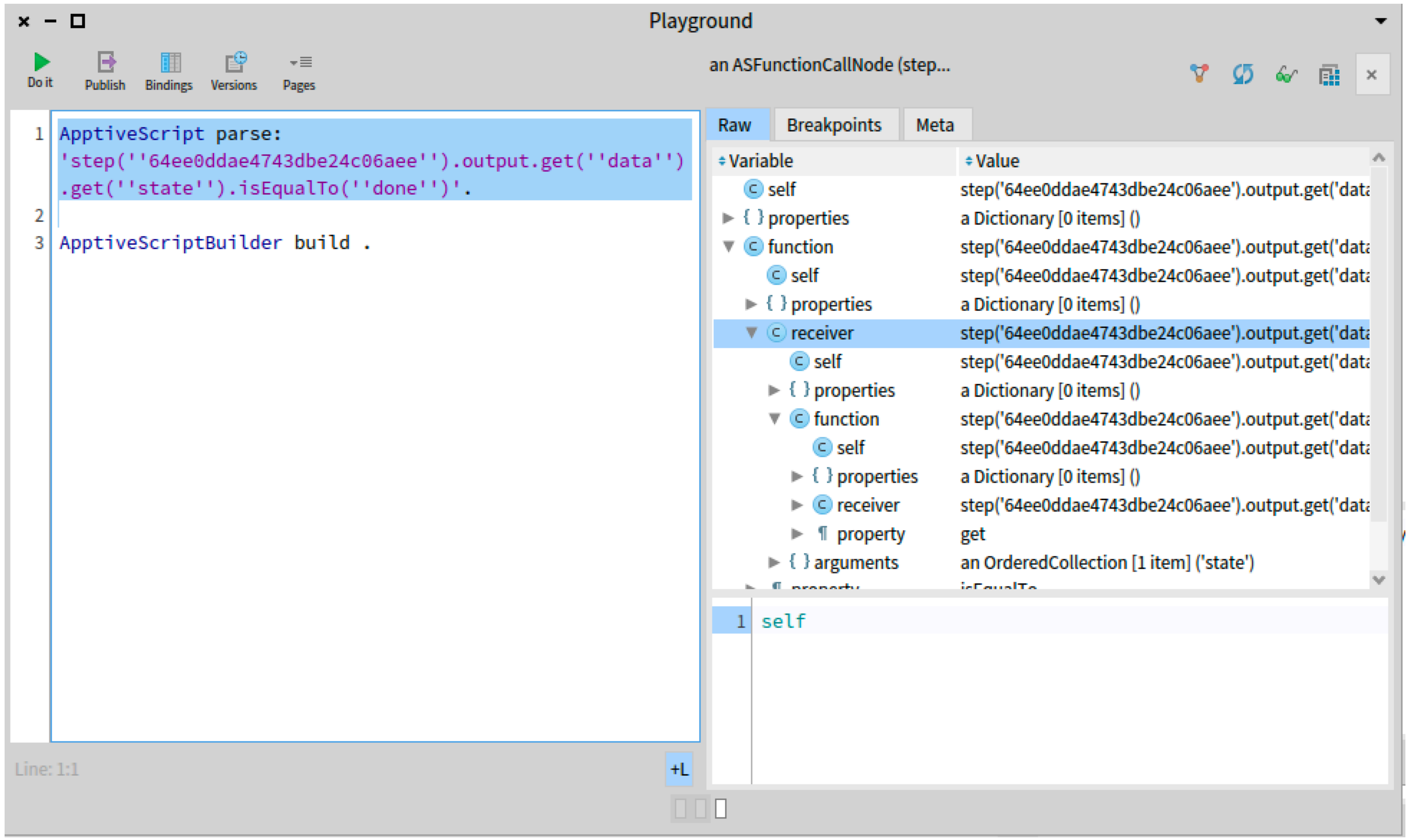
ASError !
ASNode !
ASParser !
ASProperty !
ASToken !
ASTokenizer !
ASTreeNodeVisitor !
ASCountTestVisitor !
ASFormattingVisitor !
ASScopeVisitor !
ASImportVisitor !
ASInterpreterVisitor !
ASTypeLookupVisitor !
ASTypeValidatorVisitor !

Filter...

instance side
accessing
initialization
parsing
stepping

initializeWithTokenizer:
nextToken
nextToken:
parse
parseArgumentList
parseBinaryMessage
parseBinaryMessageWith:
parseCascadedFunctionWith:
parseChainedFunction
parseExpression
parseFunction
parseNumberLiteral
parseParenthesizedExpression
parsePrimitive
parsePrimitiveIdentifier

ApptiveScript



Playground

Do it | Publish | Bindings | Versions | Pages

```
ApptiveScript parse:
'step(''64ee0ddae4743dbe24c06aee'').output.get(''data'')
.get(''state'').isEqualTo(''done'')'.

ApptiveScriptBuilder build .
```

Line: 1:1

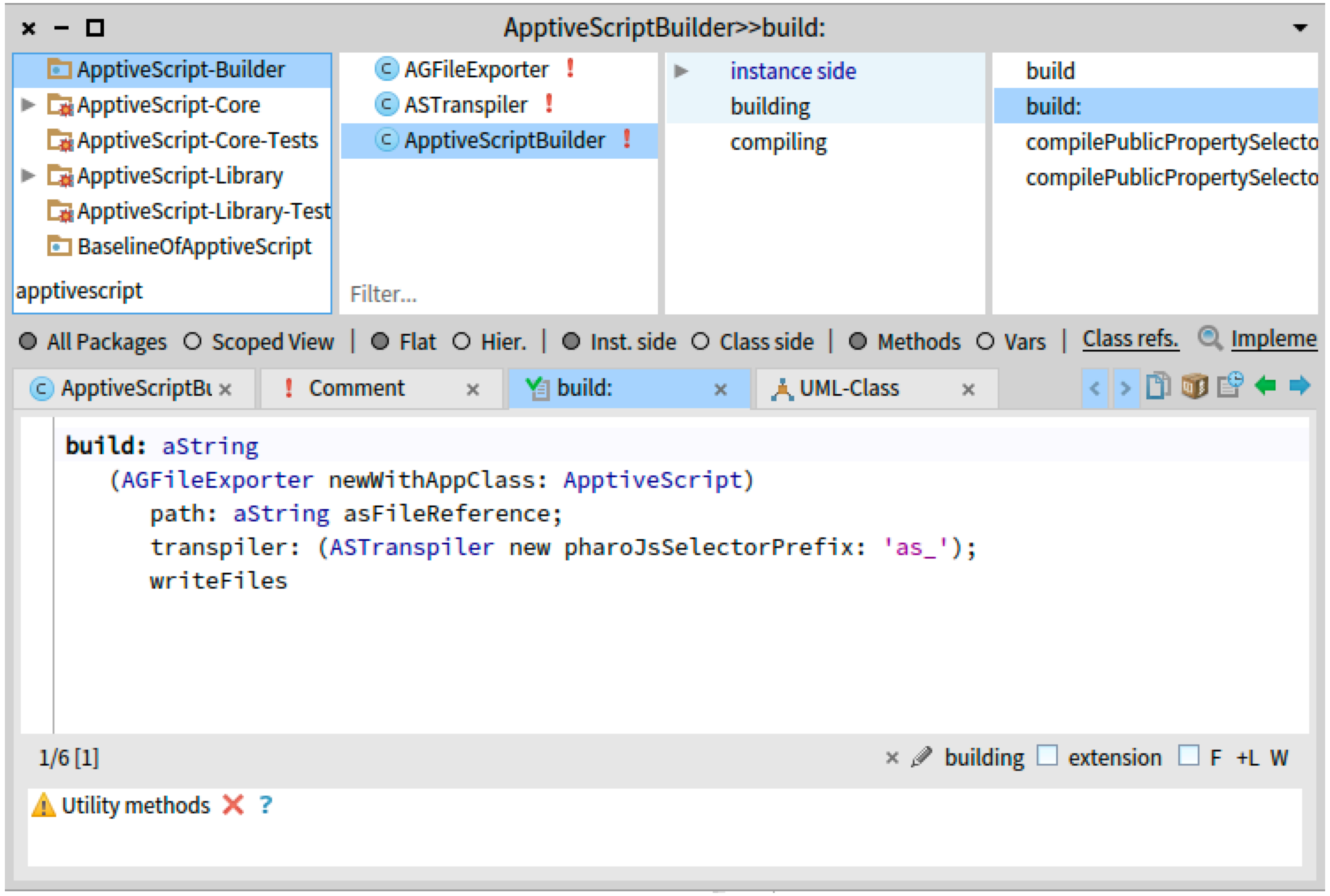
an ASFunctionCallNode (step...

Raw | Breakpoints | Meta

Variable	Value
self	step('64ee0ddae4743dbe24c06aee').output.get('data
properties	a Dictionary [0 items] ()
function	step('64ee0ddae4743dbe24c06aee').output.get('data
self	step('64ee0ddae4743dbe24c06aee').output.get('data
properties	a Dictionary [0 items] ()
receiver	step('64ee0ddae4743dbe24c06aee').output.get('data
self	step('64ee0ddae4743dbe24c06aee').output.get('data
properties	a Dictionary [0 items] ()
function	step('64ee0ddae4743dbe24c06aee').output.get('data
self	step('64ee0ddae4743dbe24c06aee').output.get('data
properties	a Dictionary [0 items] ()
receiver	step('64ee0ddae4743dbe24c06aee').output.get('data
property	get
arguments	an OrderedCollection [1 item] ('state')
property	isEqualTo

```
self
```

ApptiveScript - PharJS



```
build: aString
	(AGFileExporter newWithAppClass: ApptiveScript)
		path: aString asFileReference;
		transpiler: (ASTranspiler new pharoJsSelectorPrefix: 'as_');
		writeFiles
```

ApptiveScript on the frontend

- generated picking visual elements
- enables auto completion through type information.
- can be executed on demand
- sends syntax to the server

ApptiveScript on the backend

- is parsed to AST
- AST is stored in soil
- uses the AST as database predicate

ApptiveScript on the backend

- objects are read using an iterator (#next just as in stream)
- iterator executes the ApptiveScript AST on object
- uses the AST as predicate
- returns the result

Thank you!
Questions?