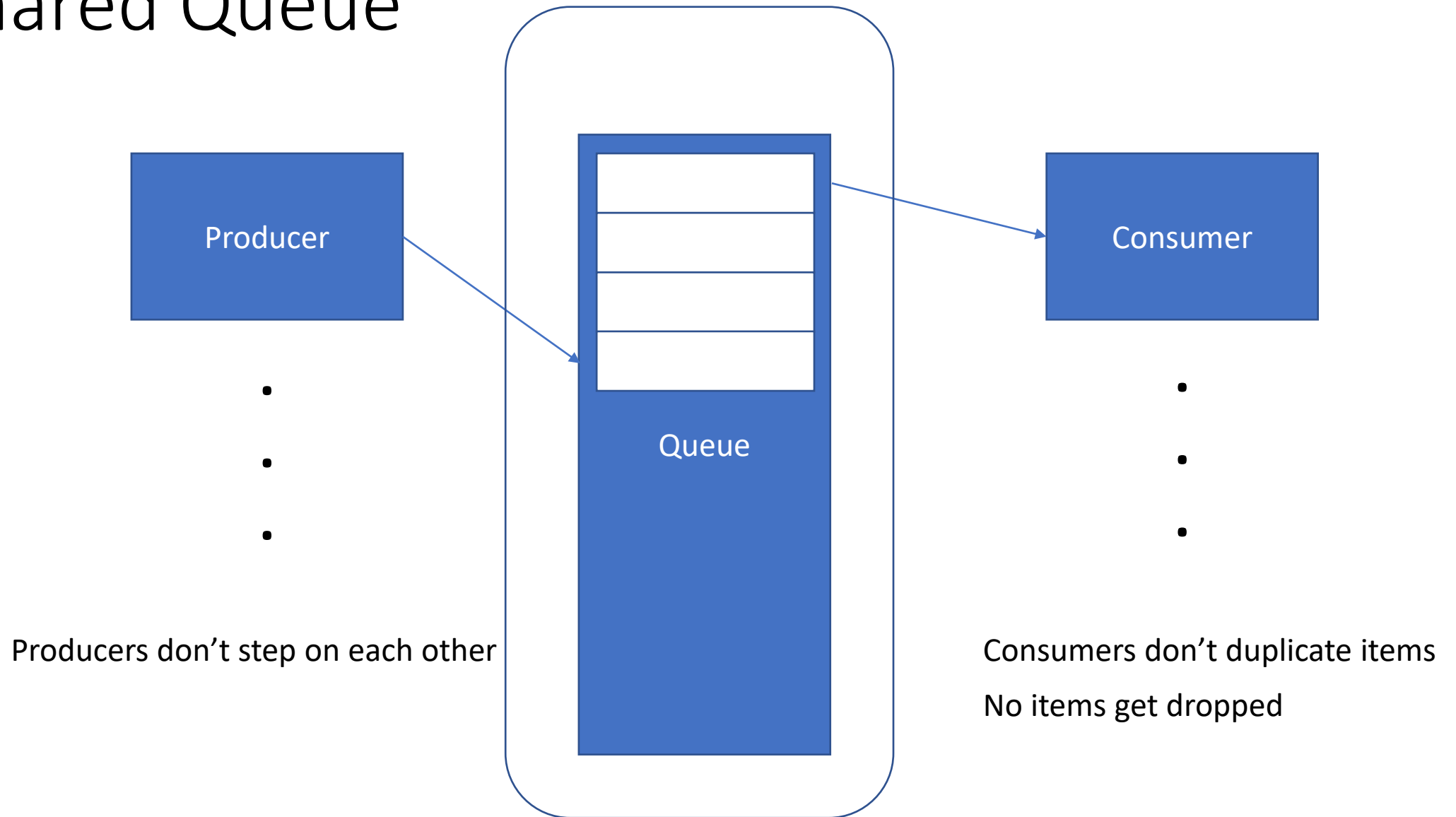


Shared Queue



REST Service – REST + Queue = `rest_queue`

- REST API basics
 - Endpoint URLs
 - Methods
 - HTTP status codes
- `rest_queue` API

REST API basics - URLs

- Collection

http://host:port/<type>s

- Single item

http://host:port/<type>s/<instance_id>

- Execute a method on an item

http://host:port/<type>s/<instance_id>/<method>

- Collection of sub-items

http://host:port/<type>s/<instance_id>/<subtype>s

- Single sub-item

http://host:port/<type>s/<instance_id>/<subtype>s/<subinstance_id>

REST API basics - Methods

Create	- POST	<code>http://host:port/<type>s</code>
Read	- GET	<code>http://host:port/<type>s/<instance_id></code>
Update	- PUT/PATCH	<code>http://host:port/<type>s/<instance_id></code>
Delete	- DELETE	<code>http://host:port/<type>s/<instance_id></code>
Execute	- POST	<code>http://host:port/<type>s/<instance_id>/<method></code>
Relate	- POST	<code>http://host:port/<type>s/<instance_id>?<relation>=<instance_id></code>
List	- GET	<code>http://host:port/<type>s</code>

HTTP status codes

- Return codes are grouped by 100's, and roughly translate to:
 - 100-199 – informational (rarely used in REST, if ever)
 - 200-299 – success
 - 300-399 – redirection
 - 400-499 – client code screwed up
 - 500-599 – server code screwed up
- REST services most commonly return:
 - 200 – OK
 - 201 – Created
 - 202 – Accepted
 - 204 – No content (ok, and no further info in the body of the response)
 - 400 – Bad request (generic error, not really helpful)
 - 401 – Unauthorized (authentication credentials not given, or invalid)
 - 403 – Forbidden (authenticated user is not permitted to perform this command)
 - 404 – Not found (no such resource)
 - 409 – Conflict (resource not in a valid state for this command)
- Handy reference for status codes: <http://httpstatuscodes.com>

rest_queue REST API

Create	- POST	<code>http://host:port/queues/<queue_name></code>
Read	- GET	<code>http://host:port/queues/<queue_name></code>
Push	- POST	<code>http://host:port/queues/<queue_name>/push</code>
Pop	- POST	<code>http://host:port/queues/<queue_name>/pop</code>
Delete	- DELETE	<code>http://host:port/queues/<queue_name></code>
List	- GET	<code>http://host:port/queues</code>

FastAPI endpoints

- From the FastAPI tutorial:

```
from fastapi import FastAPI

app = FastAPI()

@app.get("/")
async def root():
    return {"message": "Hello world"}
```

from [First Steps - FastAPI \(tiangolo.com\)](https://fastapi.tiangolo.com/tutorial/first-steps/)

- So we implement each rest_queue endpoint following this pattern