# REST & APIS

*Code 301*

# AGENDA

- Topics for Code Review
- A Conversations About REST
- RESTful APIs
- Demo of MVC and REST

# TOPICS FOR CODE REVIEW

# REST

# A DRAMATIC READING

“How I Explained REST to My Brother…”

*-https://gist.github.com/brookr/5977550*

# REST CONCEPTS

# REST CONCEPTS

- REST:
  - Representational State Transfer
- HTTP:
  - Like GPS for knowledge; applying VERB to NOUN.
- URL:
  - Points to the representation of a resource
- Web page:
  - representation of some info in HTML

# REST CONCEPTS

- REST Verbs
  - Act on a “noun” resource (hint: your Models!)
  - GET: Fetch a resource (usually to display it)
  - POST: Send info about a new resource
  - PUT: Send info about changing a resource
  - DELETE: Remove a resource, duh.

## XML-RPC/SOAP ⊘

- **Same method, varying endpoints:**
  - GET /getComment
  - GET /getComments
  - GET /addComment
  - GET /editComment
  - GET /deleteComment

## RESTFUL ROUTES

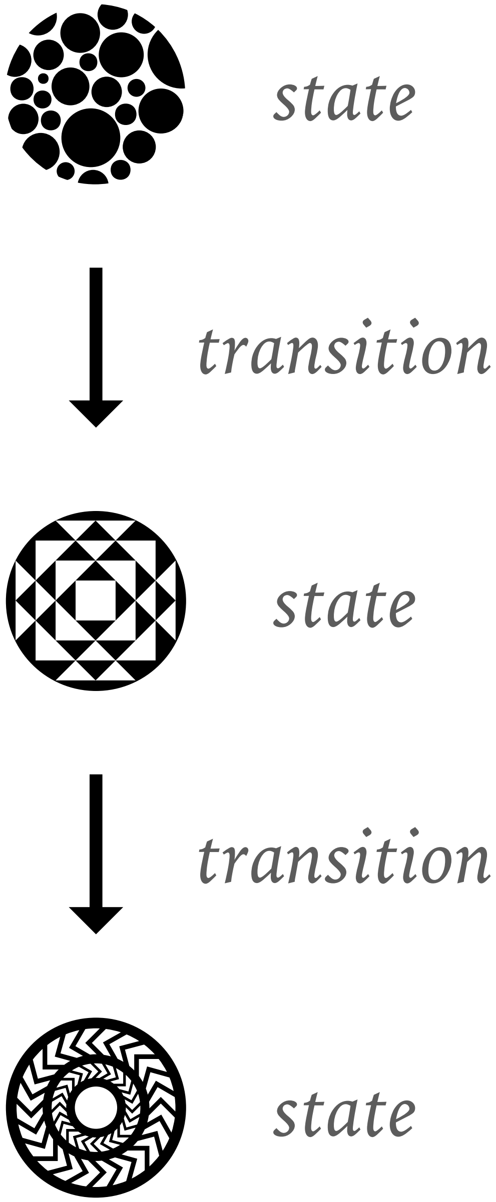
- **HTTP method + resource:**
- GET /comments/:id
- GET /comments
- POST /comments
- PUT /comments/:id
- DELETE /comments/:id
- *Resource Oriented Architecture*
- Methods address the same web-based **resource**

# RESTFUL ROUTES

- Each “endpoint” is a route, made of noun and verb.
- HTTP is **stateless**
  - Requests do not have “memory”
  - **No individual request may be assumed to know anything about any other request.**
  - All the required information representing the possible actions to take should be present in every **response**.
  - Hypermedia: “Here’s the thing you asked for, here’s what you can do next”

# APP STATE AND TRANSITION

- GET:
  - asks for the current state
- POST/PATCH/PUT/DELETE:
  - changes the state



# REST API DEMO

*https://developer.github.com/v3/*

# ASSIGNMENT DEMO