

## Learning Goals

---

- Explain CRUD in the context of web applications.
- Explain what is REST
- Create a routes reference chart for the seven RESTful routes.
- Be able to distinguish between a RESTful vs non-RESTful route.

## Vocabulary

---

- CRUD
- REST
- HTTP Verbs
- URI

## Warmup

---

Based on your intermission work:

- What does CRUD stand for?
- Look at your Task Manager controller: what are the eight different method/argument combinations contained there?
- What does each one do?

## CRUD

---

- C - Create
- R - Read
- U - Update
- D - Delete

This is a list of the operations we can perform on a *resource*.

In this context, *resource* is a generic term for something that we have in our application (e.g. tasks, people, movies). Generally, each table we create in our database holds specific instances of a particular resource. We create, read, update and destroy resources in our database.

Often, our applications will have a separate controller and model for each resource.

## REST

---

- **Representational State Transfer** (ReST) is a web architecture style
- Coined by Roy Fielding in doctoral dissertation (2000)
- REST is a simple way to organize interactions between independent systems
- A map between HTTP verb/path combinations and CRUD actions users want to perform on a resource
- RESTful applications typically treat the web like a resource

## HTTP Verbs

- GET: retrieve a resource from a url
- POST: create a new resource
- PUT: update an entire resource
- PATCH: update part of a resource
- DELETE: remove/destroy a resource

Additional information from the [W3C](#).

## URI

- **Uniform Resource Identifier**

In this context, the portion of the URL after the domain.

For the URL `https://en.wikipedia.org/wiki/Uniform_Resource_Identifier`, the URI would be `/wiki/Uniform_Resource_Identifier`.

### Turn & Talk

What is the URI for this address? `http://www.tatteredcover.com/book/9781626722934`

## Seven RESTful Routes

Pull up the controller from Task Manager. Let's see if we can create a [table](#) for seven RESTful routes allowing us to perform all CRUD actions on a Task. Include the following:

- What a user wants
- Method (verb)

- URI
- Data Prep/Manipulation
- Do we want to Redirect or Render erb
- Name of View

In small groups complete the table either on a computer or poster.

What does it do?	Verb	URI	Data Prep	Redirect or Render?	View

## Not So RESTful routes

- Do research. What is an example of a non-RESTful route?
  - `/tasks/add`

## WrapUp

---

- What is each part of CRUD?
- What is REST?
- How do we combine CRUD and REST in web applications?
- Record a version of the restful routes table in your notes

## Additional Resources

---

- [Representational State Transfer](#) on Wikipedia
- [What is REST?](#) from REST API Tutorial
- [Another Video](#)