PAGE No.
DATE / /

* API *

request
Verb
Headers
Content.

Responce
Status code
headers
Content

REST :- REpresentational State transfer.

Rest Apis parts

req	verb	URI (query string
		Headers
		request body
res		Status code
		headers
		responce body.

=) Uri Point to specific resources
- use unique identifiers
- do not use Primary keys (programmer can create errors with this insted identifire Help Programmer in readibility)

=) query string.
- use non resource Propertys
  /sites? sort = name
  / sites? Page = 1
  / sites? formate = Json

PAGE NO.
DATE / /

⇒ Verbs

Get -	Retrive a resource	select
Post -	Add a new resource	insert
Put -	update an existing "	update
Patch -	" resource with change	"
Delete -	remove resource	delete

⇒ imp. idempotency if Put fire one or 3'se then that update shoule happen if no update then doesn't give error 400 bad request (this would not varify.)

⇒ Design result
- number name
  - shouden't expose server details.
    - java, node
    - Prefere camelCasing.

⇒ Deside formate during design
Accept : app / JSON, text / xml
return : application / JSON

Common formats :

Json	: application / json
xml	: text / xml
JSONP	: application / javascript*
RSS	: application / xml + rss
ATom	: application / xml + atom

⇒ Hypermedia
- Allow result to be self-describing
- Allow programmatic navigation
- Adds complexity.

This can give some result by auto program to user

PAGE No

DATE / /

=> Paging
Http://arestame/API/sites?page=2
Page allow You to limmit the output

=> API and Security

Do you need to secure your API ?	secure it?
1. using private or personalized data	yes
2. Sending sensitive data across the 'wire'?	yes
3. using credentials of any kind?	yes
4. troying to protect against overuse of your Seravers?	yes