```
import requests, sys, json
server = "https://rest.ensembl.org" #http://ebi-cli-003:3000"


# Alignment endpoint exercise

# 1a: Get in json format the LastZ pairwise alignment for taeniopygia_guttata V
gallus_gallus for region 2:106041430-106041480:1

ext = "/alignment/region/taeniopygia_guttata/2:106041430-
106041480:1?method=LASTZ_NET;species_set=taeniopygia_guttata;species_set=gallus_
gallus"
r = requests.get(server+ext, headers={ "Content-Type" : "application/json"})
if not r.ok:
  r.raise_for_status()
  sys.exit()
decoded = r.json()
print json.dumps(decoded, indent=4, sort_keys=True)
print ("\n------->>>> Q1 Get in json format the LastZ pairwise alignment for
taeniopygia_guttata V gallus_gallus for region 2:106041430-106041480:1!!!!!
\n\n\n\n")


# 2a: Get in json format the aligned human mammal epo alignment for region
2:106040000-106040050:1

ext = "/alignment/region/human/2:106040000-
106040050:1?method=EPO;species_set_group=mammals"
r = requests.get(server+ext, headers={ "Content-Type" : "application/json"})
if not r.ok:
  r.raise_for_status()
  sys.exit()
decoded = r.json()
print json.dumps(decoded, indent=4, sort_keys=True)
print ("\n------->>>> Q2a Get in json format the aligned human mammal epo
alignment for region 2:106040000-106040050:1!!!!! \n\n\n\n")


# 2b: display only the [human, chimp, gorilla] alignment of the mammal epo
alignment for region 140000-141000.  output in phyloxml

ext = "/alignment/region/human/2:106040000-
106040050:1?method=EPO;species_set_group=mammals;display_species_set=human;displ
ay_species_set=chimp;display_species_set=gorilla"
r = requests.get(server+ext, headers={ "Content-Type" : "text/xml"})
if not r.ok:
  r.raise_for_status()
  sys.exit()
print (r.text)
print ("\n------->>>> Q2b display only the [human, chimp, gorilla] alignment of
the mammal epo alignment for region 140000-141000.  output in phyloxml!!!!!
\n\n\n\n")
```

```
# Family endpoint exercises

# Q1a: Get the information for the family with the stable id PTHR10740_SF4

ext = "/family/id/PTHR10740_SF4?"
r = requests.get(server+ext, headers={ "Content-Type" : "application/json"})
if not r.ok:
  r.raise_for_status()
  sys.exit()
decoded = r.json()
print json.dumps(decoded, indent=4, sort_keys=True)
print ("\n------->>>> Q1a Get the information for the family with the stable id PTHR10740_SF4!!!!! \n\n\n\n")


# Q1b: Get the aligned cdna sequences for only the ensembl members of the family with the stable id PTHR10740_SF4

ext = "/family/id/PTHR10740_SF4?aligned=1;sequence=cdna;member_source=ensembl"
r = requests.get(server+ext, headers={ "Content-Type" : "application/json"})
if not r.ok:
  r.raise_for_status()
  sys.exit()
decoded = r.json()
print json.dumps(decoded, indent=4, sort_keys=True)
print ("\n------->>>>Q1b Get the aligned cdna sequences for only the ensembl members of the family with the stable id PTHR10740_SF4!!!!! \n\n\n\n")


# Q2: Get the information for families predicted for the human gene ENSG00000283087. What do you notice?

ext = "/family/member/id/ENSG00000283087?"
r = requests.get(server+ext, headers={ "Content-Type" : "application/json"})
if not r.ok:
  r.raise_for_status()
  sys.exit()
decoded = r.json()
print json.dumps(decoded, indent=4, sort_keys=True)
print ("\n------->>>> Get the information for families predicted for the human gene ENSG00000283087. What do you notice?!!!!! \n\n\n\n")


# Q3: Get the information for uniprot members of families predicted for the human gene with the symbol HOXD4-001. Show no alignments and no sequences

ext = "/family/member/symbol/human/HOXD4-001?aligned=0;sequence=none;member_source=uniprot"
r = requests.get(server+ext, headers={ "Content-Type" : "application/json"})
if not r.ok:
  r.raise_for_status()
  sys.exit()
decoded = r.json()
print json.dumps(decoded, indent=4, sort_keys=True)
print ("\n------->>>> Get the information for uniprot members of families predicted for the human gene with the symbol HOXD4-001. show no alignments and no sequences!!!!! \n\n\n\n")
```

```
# Gene tree endpoint exercises

# Q1a: Get the information for the protein genetree with the stable id
ENSGT00390000003602. output should be in the orthoxml format

ext = "/genetree/id/ENSGT00390000003602?"
r = requests.get(server+ext, headers={ "Content-Type" : "text/x-orthoxml+xml"})
if not r.ok:
  r.raise_for_status()
  sys.exit()
print (r.text)
print ("\n------->>>>Q1a Get the information for the protein genetree with the
stable id ENSGT00390000003602. output should be in the orthoxml format!!!!!
\n\n\n\n")


# Q1b: Get the NcRNA gene tree with the stable id RF01168. output in phyloxml
format with sequences aligned

ext = "/genetree/id/RF01168?aligned=1;sequence=cdna"
r = requests.get(server+ext, headers={ "Content-Type" : "text/x-phyloxml+xml"})
if not r.ok:
  r.raise_for_status()
  sys.exit()
print (r.text)
print ("\n------->>>>Q1b Get the NcRNA gene tree with the stable id RF01168.
output in phyloxml format with sequences aligned!!!!! \n\n\n\n")


# Q2: Get the gene tree predicted for the gene ENSG00000189221 in full nh format

ext = "/genetree/member/id/ENSG00000189221?nh_format=full"
r = requests.get(server+ext, headers={ "Content-Type" : "text/x-nh"})
if not r.ok:
  r.raise_for_status()
  sys.exit()
print (r.text)
print ("\n------->>>>Q2 Get the gene tree predicted for the gene ENSG00000189221
in full nh format!!!!! \n\n\n\n")


# Q3: Get the gene tree predicted for the human gene with the symbol HOXD4-001
in simple nh format.

ext = "/genetree/member/symbol/human/HOXD4-001?"
r = requests.get(server+ext, headers={ "Content-Type" : "text/x-nh"})
if not r.ok:
  r.raise_for_status()
  sys.exit()
print (r.text)
print ("\n------->>>> : Get the gene tree predicted for the human gene with the
symbol HOXD4-001 in simple nh format.!!!!! \n\n\n\n")
```

```
# cafe tree endpoint exercises

# Q1: Get the cafe tree information for the genetree with the stable id
ENSGT00390000003602. output should be in the json format.

ext = "/cafe/genetree/id/ENSGT00390000003602?"
r = requests.get(server+ext, headers={ "Content-Type" : "application/json"})
if not r.ok:
  r.raise_for_status()
  sys.exit()
decoded = r.json()
print json.dumps(decoded, indent=4, sort_keys=True)
print ("\n------->>>>Q1 Get the cafe tree information for the genetree with the
stable id ENSGT00390000003602. output should be in the json format.!!!!!
\n\n\n")


# Q2: Get the cafe tree for gene tree predicted for the gene ENSG00000189221 in
nh format. Notice anything?

ext = "/cafe/genetree/member/id/ENSG00000189221?"
r = requests.get(server+ext, headers={ "Content-Type" : "text/x-nh"})
if not r.ok:
  r.raise_for_status()
  sys.exit()
print (r.text)
print ("\n------->>>>Q2 Get the cafe tree for gene tree predicted for the gene
ENSG00000189221 in nh format. Notice anything?!!!!! \n\n\n\n")


# Q3: Get the cafe tree information for the gene tree predicted for the human
gene with the symbol HOXD4-001

ext = "/cafe/genetree/member/symbol/homo_sapiens/HOXD4-001?"
r = requests.get(server+ext, headers={ "Content-Type" : "text/x-nh"})
if not r.ok:
  r.raise_for_status()
  sys.exit()
print (r.text)
print ("\n------->>>>Q3 Get the cafe tree information for the gene tree
predicted for the human gene with the symbol HOXD4-001!!!!! \n\n\n\n")
```

```
# Homology endpoint exercises

# Q1a: Get all the homologues for the human gene ENSG00000229314 in xml format

ext = "/homology/id/ENSG00000229314?"
r = requests.get(server+ext, headers={ "Content-Type" : "text/xml"})
if not r.ok:
  r.raise_for_status()
  sys.exit()
print (r.text)
print ("\n------->>>>Q1a Get all the homologues for the human gene
ENSG00000229314 in xml format!!!!! \n\n\n")


# Q1b: Return only the unaligned chimp and mouse homologs for the gene given in
1a in json format

ext =
"/homology/id/ENSG00000229314?target_species=chimp;target_species=mouse;aligned=
0"
r = requests.get(server+ext, headers={ "Content-Type" : "application/json"})
if not r.ok:
  r.raise_for_status()
  sys.exit()
decoded = r.json()
print json.dumps(decoded, indent=4, sort_keys=True)
print ("\n------->>>>Q1b Return only the unaligned chimp and mouse homologs for
the gene given in 1a in json format. !!!!! \n\n\n")


# Q2: Get all the orthologs human gene with the symbol HOXD4-001 in orthoxml
format

ext = "/homology/symbol/human/HOXD4-001?"
r = requests.get(server+ext, headers={ "Content-Type" : "text/x-orthoxml+xml"})
if not r.ok:
  r.raise_for_status()
  sys.exit()
print (r.text)
print ("\n------->>>>Q2 Get all the orthologs human gene with the symbol HOXD4-
001 in orthoxml format. !!!!! \n\n\n")
```