```javascript
// Install xml2js library
// npm install xml2js

//import the library.
const xml2js = require('xml2js');

//create instance of xml2js.
//we have parser method to parse xml.

const parser = new xml2js.Parser({ explicitArray: false });
```

```javascript
describe("XML Parsing",()⇒{

    //xml has to be in one line else we get compile error.
    //use multi to single line converter.

    const xmlPayload="<Pet>      <id>0</id>      <Category>
<id>0</id>          <name>Dog</name>      </Category>
<name>Jimmy</name>      <photoUrls>
<photoUrl>string</photoUrl>      </photoUrls>
<tags>          <Tag>              <id>0</id>
<name>string</name>          </Tag>      </tags>
<status>available</status> </Pet>"
    let petid=null;

    before("Creating new PET",()⇒{
        cy.request({
                method: 'POST',
                url: 'https://petstore.swagger.io/v2/pet',
                body:xmlPayload,
                //content type is for sending request.
                //accept for telling what type of response it is.
                headers:{'Content-Type':'application/xml',
                         'accept':'application/xml'
                        }
        }).then((response) ⇒{
                expect(response.status).to.eq(200);
                //parse string converts xml to json object.
                //error if any will be stored in err.
                //proper response if any will be stored in result.
                parser.parseString(response.body, (err,result) ⇒ {
                    petid=result.Pet.id;

                })
            });
        });
```

```javascript
it("Fetching Pet data-parsing xml response",()=>{
            cy.request({
                    method: 'GET',
                    url: "https://petstore.swagger.io/v2/pet/"+petid,
                    //since no body, so content type not mandatory to write.
                     headers:{'accept':'application/xml'
                        }
            }).then((response) =>{
                    expect(response.status).to.eq(200);
                    parser.parseString(response.body, (err,result) => {
                        expect(result.Pet.id).equal(petid);
                        expect(result.Pet.name).equal("Jimmy")

                        })
                    });
            });

})
```