New login protocol:

1. Client requests salt from the server
2. Server responds with the salt
3. Client concatenates the salt to the end of the plain text password
4. Client hashes the salted string 1023 times with SHA256
5. Client sends the hash to the server
6. Server hashes the password once and then validates it
    - Valid Password:
        1. Server responds with a token and an HTTP 200 status code (see below)
        2. The client adds this token to all requests that modify data (see below)
    - Invalid Password:
        1. Server responds with an HTTP 400 Invalid password status code
        2. The user is asked to enter the password again

Tokens:

When there is a successful login attempt the server creates a token by string concatenating 3 64-bit integers, effectively creating a 192-bit token. The server stores this token for the duration of its runtime unless it is cleared by a password change. The user sends this token in any requests that modify data (PUT, POST, DELETE requests) with the 'token' parameter in the query string. This should be more than secure enough because guessing a 192-bit number is basically impossible (The numbers are generated with RNGCryptoServiceProvider in c# which should be crytographically secure), and brute forcing all 192-bit numbers is also basically impossible.

Server Protocol

- **REQUEST_TYPE** -> HandlerMethod
    - Example URI
        - All Options
        - **Required Option**
        - *Option in body* (See Body Format at bottom of page)
    - Results
    - Notes

General Results:

- **401 Unauthorized (Bad Token)** if a secure request has a bad token
- **401 Unauthorized (Secure Connection Required)** if a type that requires HTTPS is being called from an insecure connection while the server is required secure connections. This applies to most DELETE, POST, and PUT requests, with some exceptions (Like CREATE_FEEDBACK).

HEAD -> HandleHeadRequest

- **CHECK_TOKEN** -> HandleCheckToken
  - /?Type=CheckToken&Token=3902309309034...
    - **Token**: Auth Token to check
  - Results
    - **200 OK** if the token is valid
    - **401 Unauthorized** if the token is not valid

- ***CLEAR_TOKEN*** -> HandleClearToken
  - /?Type=ClearToken&Token=389489493493...
    - **Token**: Auth Token to clear
  - Results
    - **200 OK** always
  - Clears a token so it can no longer be used

GET -> HandleGetRequest

- ***DOQ_TYPE*** -> HandleGetDoq
  - /?Type=Doq&Guid=38338FA3893C-83943...&Count=15
    - **Guid**: Guid of the doq to get
    - Count: Client's change counter for the doq
  - Results
    - **200 OK** If the doq is found, body contains the doq serialized to XML
    - **304 Not Modified** If the provided change counter matches the server's change counter
    - **404 Not Found** If the doq is not found
  - Might not need to be used

- **ALL_ARTWORKS** -> HandleGetFolder
  - /?Type=Artworks
    - Count: Client's change counter for artworks
  - Results
    - **200 OK** If the change counter does not match the server's change counter, body contains the folder serialized to XML
    - **304 Not Modified** See DOQ_TYPE

- **ALL_TOURS** -> HandleGetFolder
  - /?Type=Tours
    - Count: Client's change counter for tours
  - Results
    - **200 OK** See ALL_ARTWORKS
    - **304 Not Modified** See DOQ_TYPE

- **ALL_EXHIBITIONS** -> HandleGetFolder
  - /?Type=Exhibitions
    - Count: Client's change counter for exhibitions
  - Results
    - **200 OK** See ALL_ARTWORKS
    - **304 Not Modified** See DOQ_TYPE

- **ALL_ASSOCIATED_MEDIA** -> HandleGetFolder
  - /?Type=AllAssociatedMedia
    - Count: Client's change counter for associated media
  - Results
    - **200 OK** See ALL_ARTWORKS
    - **304 Not Modified** See DOQ_TYPE

- **ALL_FEEDBACK** -> HandleGetFolder
  - /?Type=Feedback
    - Count: Client's change counter for feedback
  - Results
    - **200 OK** See ALL_ARTWORKS
    - **304 Not Modified** See DOQ_TYPE

- **GET_SALT** -> HandleGetSalt
  - /?Type=Salt
  - Results
    - **200 OK** Always
  - Returns the salt for the password which is basically publicly available if someone figures out this request, but that shouldn't really matter, the point is that a pre-computed rainbow table doesn't exist for our salt provided its a big enough salt (it is). Additionally, our hashing technique makes rainbow table construction take much longer.

- **AUTH** -> HandleAuth
  - /?Type=Auth&Hash=d1e35505fed93e6d52f069ab87c9b764bd864a69cf286bd0167a9e591eab941a
    - **Hash**: Hash of the password after 1023 rounds of SHA256
  - Results
    - **200 OK** If the password is valid, body contains a new token
    - **401 Not Authorized** If the password is not valid

- **MAIN** -> HandleGetDoq
  - /?Type=Main
    - Count: Client's change counter for main
  - Results
    - **200 OK** See DOQ_TYPE

        - **304 Not Modified** See DOQ_TYPE

- **ASSOCIATED_MEDIA** -> HandleGetAssociatedMedia
    - /?Type=AssociatedMedia&Guid=3909034FA83AC-23...
        - **Guid**: Guid of the doq to get associated media for
        - Count: Client's change counter for main
    - Results
        - **200 OK** See DOQ_TYPE
        - **304 Not Modified** See DOQ_TYPE

- **CHECK_VERSION** -> HandleGetVersion
    - /?Type=CheckVersion
    - Results
        - **200 OK** Returns the server version as the status text and the main doq's identifier in the response body

- **LINQ_DATA** -> HandleGetLinq
    - /?Type=Linq&Guid1=89389393...&Guid2=38993893...
        - **Guid1**: Guid of one doq in the linq
        - **Guid2**: Guid of the other doq in the linq
        - Count: Client's change counter for main
    - Results
        - **200 OK** See DOQ_TYPE
        - **304 Not Modified** See DOQ_TYPE
        - **404 Not Found** If the linq is not found

- **ARTWORKS_IN_EXHIBITION** -> HandleArtworksInExhibition
    - /?Type=ArtworksIn&Guid=839848934...
        - **Guid**: Guid of the exhibition to get artworks for
        - Count: Client's change counter for artworks in the exhibition
    - Results
        - **200 OK** See DOQ_TYPE
        - **304 Not Modified** See DOQ_TYPE
        - **404 Not Found** If the guid is not found or not an exhibition

DELETE -> HandleDeleteRequest

- **DOQ_TYPE** -> HandleDeleteDoq
    - /?Type=Doq&Guid=3909034FA83AC-23...&Force=true
        - **Guid**: Guid of the doq to delete
        - Count: Client's change counter for the doq
        - Force: boolean, whether or not to always delete
    - Results
        - **200 OK** The doq has been deleted

        - **404 Not Found** The doq was not found
        - **409 Conflict** The doq was not deleted because the change counter did not match the server's change counter
    - The provided change counter must match the server's change counter unless force is true, in which case the doq will always be deleted. The idea is to prevent someone from deleted a doq if someone else made a change to it between refreshes.

- **LINQ_TYPE** -> HandleDeleteLinq
    - /?Type=Linq&Guid=3909034FA83AC-23...
        - **Guid**: Guid of the linqto delete
        - Count: Client's change counter for the linq
        - Force: boolean, whether or not to always delete
    - Results
        - **200 OK** The linq has been deleted
        - **404 Not Found** The linq was not found
        - **409 Conflict** The linq was not deleted because the change counter did not match the server's change counter
    - The provided change counter must match the server's change counter unless force is true, in which case the linq will always be deleted. The idea is to prevent someone from deleted a linq if someone else made a change to it between refreshes.

POST -> HandlePostRequest

- **FILE_UPLOAD** -> HandleFileUpload
    - /?Type=FileUpload&Extension=png&Client=Windows
        - **Extension**: Extension of the file being uploaded
        - **Client**: Temporary hack because multipart upload might not work, should always be set to Windows
    - Results
        - **200 OK** The file has been uploaded, body contains the url
        - **415 Unsupported Media Type** The file was not uploaded because the extension was not valid

- **FILE_UPLOAD_DEEPZOOM** -> HandleFileUploadDeepzoom
    - /?Type=FileUpload&Extension=png&Client=Windows&ReturnDoq=true
        - **Extension**: Extension of the file being uploaded
        - **Client**: Temporary hack because multipart upload might not work, should always be set to Windows
        - ReturnDoq: boolean, whether or not the response should contain the created doq serialized to XML
    - Results

        - **200 OK** The file has been uploaded, body contains the serialized doq if ReturnDoq is true
        - **415 Unsupported Media Type** The file was not uploaded because the extension was not valid

- **FILE_UPLOAD_DATAURL** -> HandleFileUploadDeepzoom
    - /?Type=FileUploadDataURL
    - Results
        - **200 OK** The file has been uploaded, body contains the url
        - **400 Bad Request** If no data is sent

- **CHANGE_PASSWORD** -> HandleChangePassword
    - /?Type=ChangePassword&OldHash=cba06b5736faf67e54b07b561eae94395e774c517a7d910a54369e1263ccfbd4&NewPass=P4ssW0rd!
        - **OldHash**: The old password hashed 1023 times with SHA256
        - **NewPass**: The new password in plain text (definitely requires HTTPS for any security)
    - Results
        - **200 OK** The password has been changed, response contains a new token
        - **401 Unauthorized** The OldHash is incorrect
        - **403 Forbidden** The server is not allowing password change requests

- **CHANGE_TOUR** -> HandleChangeTour
    - /?Type=ChangeTour&Guid=2080893...&Name=Published Tour&Private=false&Force=true
        - **Guid**: guid of the tour to change
        - Name: New name
        - Thumbnail: New thumbnail URL
        - Private: New private state
        - RelatedArtworks: JSON of related artworks
        - Count: Client's change counter
        - Force: boolean, whether or not to always delete
        - *Description*: New description
        - *Content*: New content
    - Results
        - **200 OK** The tour was successfully changed
        - **404 Not Found** The tour was not found
        - **409 Conflict** The tour was not changed because the change counter did not match the server's change counter and force was not set to true

- **CHANGE_ARTWORK** -> HandleChangeArtwork
    - /?Type=ChangeArtwork&Guid=2080893...&Artist=JJStory&&Preview=/Images/New.jpg
        - **Guid**: guid of the artwork to change

        - Name: New name
        - Title: New title (is this used??)
        - Artist: New artist name
        - Year: New year
        - Preview: new preview image URL
        - Thumbnail: New thumbnail URL
        - Deepzoom: New Deepzoom URL
        - Count: Client's change counter
        - Force: boolean, whether or not to always delete
        - *InfoFields*: JSON of custom info fields
        - *Description*: New description
        - *Location*: New location info
        - *AddIDs*: csv of IDs to add as associated media
        - *RemoveIDs*: csv of IDs to remove as associated media
    - Results
        - **200 OK** See CHANGE_TOUR
        - **404 Not Found** See CHANGE_TOUR
        - **409 Conflict** See CHANGE_TOUR

- **CHANGE_EXHIBITION** -> HandleChangeExhibition
    - /?Type=ChangeExhibition&Guid=2080893...&Sub1=Subheading&Background=/Images/TagUnicorn.jpg
        - **Guid**: guid of the exhibition to change
        - Name: New name
        - Sub1: New subheading 1
        - Sub2: New subheading 2
        - Background: New BG Image URL
        - Img1: Desc image 1 URL
        - Img2: Desc image 2 URL (not used anymore)
        - Private: New private state
        - Count: Client's change counter
        - Force: boolean, whether or not to always delete
        - *Description*: New description
        - *AddIDs*: csv of artwork IDs to add to the exhibition
        - *RemoveIDs*: csv of artwork IDs to remove from the exhibition
    - Results
        - **200 OK** See CHANGE_TOUR
        - **404 Not Found** See CHANGE_TOUR
        - **409 Conflict** See CHANGE_TOUR

- **CHANGE_ASSOCIATED_MEDIA** -> HandleChangeAssociatedMedia
    - /?Type=ChangeAssociatedMedia&Guid=2080893...&LinqTo=389983...&X=47.3&Y=52.2&Type=Hotspot&Duration=200
        - **Guid**: guid of the associated media to change

        - Name: New name
        - ContentType: New content type
        - Duration: new duration
        - Source: new source
        - LinqTo: If the user wants to edit a linq they can supply the doq this media is linqed to
        - X: The new X position of the linq (requires LinqTo)
        - Y: The new Y position of the linq (requires LinqTo)
        - LinqType: The new type of the linq (requires LinqTo)
        - Count: Client's change counter
        - Force: boolean, whether or not to always delete
        - *Description*: New description
        - *AddIDs*: csv of IDs of artwork to add a linq to
        - *RemoveIDs*: csv of IDs of artwork to remove a linq to
    - Results
        - **200 OK** See CHANGE_TOUR
        - **404 Not Found** See CHANGE_TOUR
        - **409 Conflict** See CHANGE_TOUR

- **MAIN** -> HandleChangeMain
    - /?Type=Main&Guid=2080893...&Location=Providence&IconColor=#ABCDEF
        - Name: New name
        - OverlayColor: New overlay color
        - OverlayTrans: new overlay transparecny
        - Location: New location
        - Background: New BG Image
        - Icon: New Icon image
        - IconColor: New icon color
        - Count: Client's change counter
        - Force: boolean, whether or not to always delete
        - *Info*: New info
    - Results
        - **200 OK** See CHANGE_TOUR
        - **404 Not Found** See CHANGE_TOUR
        - **409 Conflict** See CHANGE_TOUR

PUT -> HandlePutRequest

- **CREATE_DOQ_EXHIBITION** -> HandlePutExhibitionRequest
    - /?Type=CreateExhibition
        - Name: New name (Default: Exhibition)
        - Sub1: New subheading 1 (Default: First Subheading)
        - Sub2: New subheading 2 (Default: Second Subheading)
        - Background: New BG Image URL (Default: /Images/default.jpg)

        - Img1: Desc image 1 URL (Default: /Images/default.jpg)
        - Img2: Desc image 2 URL (not used anymore)
        - Private: New private state (Default: true)
        - *Description*: New description (Default: Description)
        - *AddIDs*: csv of artwork IDs to add to the exhibition
        - ReturnDoq: If true the created doq is serialized and returned
    - Results
        - **200 OK** contains the serialized doq if ReturnDoq is true

- **CREATE_DOQ_ARTWORK** -> HandlePutArtworkRequest
    - /?Type=CreateArtwork
        - Name: New name
        - Title: New title (is this used??)
        - Artist: New artist name
        - Year: New year
        - Preview: new preview image URL
        - Thumbnail: New thumbnail URL
        - Deepzoom: New Deepzoom URL
        - InfoFields: JSON of custom info fields
        - *Description*: New description
        - *Location*: New location info
        - *AddIDs*: csv of IDs to add as associated media
        - ReturnDoq: If true the created doq is serialized and returned
    - Results
        - **200 OK** contains the serialized doq if ReturnDoq is true

- **CREATE_DOQ_TOUR** -> HandlePutTourRequest
    - /?Type=CreateTour
        - Name: New name
        - Thumbnail: New thumbnail URL
        - RelatedArtworks: JSON of related artworks
        - Private: New private state
        - *Description*: New description
        - *Content*: New content
        - ReturnDoq: If true the created doq is serialized and returned
    - Results
        - **200 OK** contains the serialized doq if ReturnDoq is true

- **CREATE_HOTSPOT** -> HandlePutHotspotRequest
    - /?Type=CreateHotspot
        - Name: New name
        - ContentType: New content type
        - Duration: new duration
        - Source: new source

            - LinqTo: If the user wants to create a linq they can specify the doq id to linq to
            - X: The new X position of the linq (requires LinqTo)
            - Y: The new Y position of the linq (requires LinqTo)
            - LinqType: The new type of the linq (requires LinqTo)
            - *Description*: New description
            - ReturnDoq: If true the created doq is serialized and returned
        - Results
            - **200 OK** contains the serialized doq if ReturnDoq is true

- **CREATE_DOQ_FEEDBACK** -> HandlePutFeedbackRequest
    - /?Type=CreateFeedback
        - SourceID: Source ID
        - SourceType: Source type
        - *Description*: New description
        - ReturnDoq: If true the created doq is serialized and returned
    - Results
        - **200 OK** contains the serialized doq if ReturnDoq is true

Body Format:

If a request is supposed to have a body it is formatted as follows:

```
Boundary:<Boundary>\r\n

<Boundary>\r\n

Name:Value\r\n

<Boudnary>\r\n

Name:Value
```

<Boundary> should be a string that is not found anywhere in the keys/values being supplied. A key should not contain a colon.

Pseudocode for a boundary-finding algorithm could look as follows:

```
boundary=""

found=false

do:
```

```
  boundary=boundary + "-"
  for name in names:
    if name contains boundary:
      found=true
      break
  for key in keys:
    if key contains boundary:
      found=true
      break
while found==true
```