Asperguide Backend - HTTP Request Flow Through System Layers

Participants:
- Client
- Uvicorn ASGI Server
- FastAPI Application
- PathManager Route Dispatcher
- EndpointManager Business Logic
- OAuthAuthentication Token Validator
- BoilerplateIncoming Request Validator
- SQL Database Layer
- Redis Cache Layer
- BoilerplateResponses Response Builder
- ServerHeaders CORS/Security
- note

Request Reception

- Client → Uvicorn ASGI Server: HTTP Request (GET /api/users/123)
- Uvicorn → FastAPI: Forward ASGI Request
- FastAPI → PathManager: Route Lookup (/api/users/{id})
- PathManager → EndpointManager: Dispatch to Handler get_user_by_id()

Authentication & Authorization

- EndpointManager → OAuthAuthentication: validate_token(headers)

alt [Token in Redis cache]
- OAuthAuthentication → Redis: get_cached_token()
- Redis → OAuthAuthentication: token_data

[Token not cached]
- OAuthAuthentication → OAuthAuthentication: decode_jwt(token)
- OAuthAuthentication → Redis: cache_token(token_data)

alt [Valid Token]
- OAuthAuthentication → EndpointManager: user_context

[Invalid Token]
- OAuthAuthentication → EndpointManager: AuthenticationError
- EndpointManager → BoilerplateResponses: error_response(401)
- BoilerplateResponses → EndpointManager: JSON Response
- EndpointManager → FastAPI: 401 Unauthorized
- FastAPI → Uvicorn: HTTP Response
- FastAPI → ServerHeaders: apply_headers(response)
- ServerHeaders → FastAPI: response + CORS
- FastAPI → Uvicorn: ASGI Response
- Uvicorn → Client: HTTP 401
- *Early return on auth failure*

Request Validation

- EndpointManager → BoilerplateIncoming: validate_request(params)

alt [Invalid Parameters]
- BoilerplateIncoming → EndpointManager: ValidationError
- EndpointManager → BoilerplateResponses: error_response(400)
- BoilerplateResponses → EndpointManager: JSON Response
- EndpointManager → FastAPI: 400 Bad Request
- FastAPI → Uvicorn: HTTP Response
- FastAPI → ServerHeaders: apply_headers(response)
- ServerHeaders → FastAPI: response + CORS
- FastAPI → Uvicorn: ASGI Response
- Uvicorn → Client: HTTP 400
- *Early return on validation failure*

[Valid Parameters]
- BoilerplateIncoming → EndpointManager: validated_data

Data Access Layer

- EndpointManager → SQL: get_user(user_id)
- SQL → Redis: check_cache(query_key)

alt [Cache Hit]
- Redis → SQL: cached_result

[Cache Miss]
- SQL → SQL: execute_query()
- SQL → Redis: cache_result(query_key, result)

alt [User Found]
- SQL → EndpointManager: user_object

[User Not Found]
- SQL → EndpointManager: None
- EndpointManager → BoilerplateResponses: error_response(404)
- BoilerplateResponses → EndpointManager: JSON Response
- EndpointManager → FastAPI: 404 Not Found
- FastAPI → Uvicorn: HTTP Response
- FastAPI → ServerHeaders: apply_headers(response)
- ServerHeaders → FastAPI: response + CORS
- FastAPI → Uvicorn: ASGI Response
- Uvicorn → Client: HTTP 404
- *Early return on not found*

Response Building

- EndpointManager → BoilerplateResponses: success_response(user_object)
- BoilerplateResponses → BoilerplateResponses: serialize_data()
- BoilerplateResponses → BoilerplateResponses: add_metadata()
- BoilerplateResponses → EndpointManager: JSON Response (200)
- EndpointManager → FastAPI: 200 OK + JSON
- FastAPI → Uvicorn: HTTP Response

Response Headers & CORS

- FastAPI → ServerHeaders: apply_headers(response)
- ServerHeaders → ServerHeaders: add_cors_headers()
- ServerHeaders → ServerHeaders: add_security_headers()
- ServerHeaders → FastAPI: response + headers

Response Delivery

- FastAPI → Uvicorn: ASGI Response
- Uvicorn → Client: HTTP 200 OK + JSON Body + CORS Headers

Notes

Data Flow Pattern:
1. Authentication first
2. Request validation
3. Cache check
4. Database query (if needed)
5. Response building

Performance Optimization:
- JWT tokens cached in Redis
- Database results cached
- Early returns on errors

Response Standardization:
- Consistent JSON structure
- Standard error codes
- CORS headers on all responses