### **Get all groups (optional filters)**

|  |  |
| --- | --- |
| **URL** | /groups |
| **Method** | GET |
| **URL Params** | **Optional:**  query=string  type=GroupType  userId=number  order=GroupOrder  filter=GroupFilter  includeDiscussion=Boolean  offset=integer  limit=integer |
| **Data Params** | None |
| **Success Response** | **Code:** 200 OK  **Content:** [  {  “id”:[integer]  “key”:[string],  “type”:[GroupType],  “title”:[string],  “description”:[string],  “pictureUrl”:[string],  “avatarUrl”:[string],  “usersCount”:[integer],  “meGroupInfo”: {  “userGroupId”:[integer],  “status”:[GroupMemberStatus]  },  “levelToEditor”:[integer],  “visibilityStatus”:[ApiVisibilityStatus],  “userDiscussionInfo”: {  “lastTime”:[long],  “newCommentsCount”:[integer],  “isBookmarked”:[boolean]  },  “commentsCount”:[integer]  },  {...},  ...  ] |
| **Error Response** | **Code:** 40x  **Content:** {  “code”:[ErrorInfo],  “message”:[string],  “developerMessage”:[string]  } |
| **Notes** | includeDiscussion == true, тогда значения в userDiscussionInfo(минимальная информация о дискусии) и commentsCount будут не пустые.  public enum GroupType {  UNKNOWN,  CITY,  COMPANY,  COLLEGE,  TECH,  SPECIAL  }  public enum GroupOrder {  UNKNOWN,  NAME,  MEMBERS  }  public enum GroupFilter {  UNKNOWN,  ALL,  MY  }  public enum GroupMemberStatus {  UNKNOWN,  CANDIDATE,  INVITEE,  MEMBER,  EDITOR,  MODERATOR,  ADMINISTRATOR,  BANNED,  }  public enum ApiVisibilityStatus {  UNKNOWN,  RESTRICTED,  PUBLIC,  PROTECTED,  PRIVATE,  DISABLED,  DELETED,  } |
| **Permissions** | All |

### 

### 

### **Get groups count**

|  |  |
| --- | --- |
| **URL** | /groups/count |
| **Method** | GET |
| **URL Params** | **Optional:**  query=string  type=GroupType  userId=number  filter=GroupFilter |
| **Data Params** | None |
| **Success Response** | **Code:** 200 OK  **Content:**[integer] |
| **Error Response** | **Code:** 40x  **Content:** {  “code”:[ErrorInfo],  “message”:[string],  “developerMessage”:[string]  } |
| **Notes** |  |
| **Permissions** | All |

### 

### 

### **Get group**

|  |  |
| --- | --- |
| **URL** | /groups/group{kid} |
| **Method** | GET |
| **URL Params** | **Required:**  kid=string |
| **Data Params** | None |
| **Success Response** | **Code:** 200 OK  **Content:** {  “id”:[integer]  “key”:[string],  “type”:[GroupType],  “title”:[string],  “description”:[string],  “pictureUrl”:[string],  “avatarUrl”:[string],  “usersCount”:[integer],  “meGroupInfo”: {  “userGroupId”:[integer],  “status”:[GroupMemberStatus]  },  “levelToEditor”:[integer],  “visibilityStatus”:[ApiVisibilityStatus],  “userDiscussionInfo”: {  “lastTime”:[long],  “newCommentsCount”:[integer],  “isBookmarked”:[boolean]  },  “commentsCount”:[integer]  } |
| **Error Response** | **Code:** 40x  **Content:** {  “code”:[ErrorInfo],  “message”:[string],  “developerMessage”:[string]  } |
| **Notes** |  |
| **Permissions** | All |

### 

### 

### **Get groups where user has one of roles**

|  |  |
| --- | --- |
| **URL** | /groups/my |
| **Method** | GET |
| **URL Params** | **Optional:**  statuses= GroupMemberStatus |
| **Data Params** | None |
| **Success Response** | **Code:** 200 OK  **Content:** [  {  “id”:[integer],  “key”:[string],  “title”:[string],  “description”:[string],  “type”:[GroupType],  “avatarUrl”:[string],  “pictureUrl”:[string]  }  {...},  ...  ] |
| **Error Response** | **Code:** 40x  **Content:** {  “code”:[ErrorInfo],  “message”:[string],  “developerMessage”:[string]  } |
| **Notes** | Если роль (statuses) не передана, вернуться все группы, в которых состоит пользователь (MEMBER, EDITOR, MODERATOR, ADMINISTRATOR)  Enum GroupMemberStatus {  UNKNOWN,  CANDIDATE,  INVITEE,  MEMBER,  EDITOR,  MODERATOR,  ADMINISTRATOR,  BANNED  } |
| **Permissions** | All |

### **Get all members for current group**

|  |  |
| --- | --- |
| **URL** | /groups/group{kid}/members |
| **Method** | GET |
| **URL Params** | **Required:**  kid=string  **Optional:**  status=GroupMemberStatus  query=string //user’s displayName  limit=integer  offset=integer |
| **Data Params** | None |
| **Success Response** | **Code:** 200 OK  **Content:** [  {  “id”:[integer],  “status”:[GroupMemberStatus],  “position”:[integer],  “rating”:[integer],  “userInfo”: {  “userId”:[integer],  “displayName”:[string],  “pictureUrl”:[string],  “level”:[integer],  “solvedTasks”:[integer],  }  },  {...},  ...  ] |
| **Error Response** | **Code:** 40x  **Content:** {  “code”:[ErrorInfo],  “message”:[string],  “developerMessage”:[string]  } |
| **Notes** | Order by rating ASC |
| **Permissions** | All |

### 

### **Get members count for current group**

|  |  |
| --- | --- |
| **URL** | /groups/group{kid}/members/count |
| **Method** | GET |
| **URL Params** | **Required:**  kid=string  **Optional:**  status=GroupMemberStatus  query=string |
| **Data Params** | None |
| **Success Response** | **Code:** 200 OK  **Content:**[integer] |
| **Error Response** | **Code:** 40x  **Content:** {  “code”:[ErrorInfo],  “message”:[string],  “developerMessage”:[string]  } |
| **Notes** |  |
| **Permissions** | All |

### 

### 

### **Get member from current group**

|  |  |
| --- | --- |
| **URL** | /groups/group{kid}/members/member{id} |
| **Method** | GET |
| **URL Params** | **Required:**  kid=string  id=integer |
| **Data Params** | None |
| **Success Response** | **Code:** 200 OK  **Content:** {  “id”:[integer],  “status”:[GroupMemberStatus],  “position”:[integer],  “rating”:[integer],  “userInfo”: {  “userId”:[integer],  “displayName”:[string],  “pictureUrl”:[string],  “level”:[integer],  “solvedTasks”:[integer],  } |
| **Error Response** | **Code:** 40x  **Content:** {  “code”:[ErrorInfo],  “message”:[string],  “developerMessage”:[string]  } |
| **Notes** |  |
| **Permissions** | All |

### 

### 

### **Get all posts**

|  |  |
| --- | --- |
| **URL** | /posts |
| **Method** | GET |
| **URL Params** | **Optional:**  query=String  type=PostType  groupKid=string  groupType=GroupType  ownerId=number  order=PostOrder  filter=PostFilter  limit=number  offset=number |
| **Data Params** | None |
| **Success Response** | **Code:** 200 OK  **Content:** [  {  “id”:[integer],  “key”:[integer],  “title”:[string],  “description”:[string],  “content”:[string],  “pictureUrl”:[string],  “type”:[PostType],  “views”:[integer],  “commentsCount”:[integer],  “createdTime”:[Long],  “updatedTime”:[Long],  “rating”:[Double],  “ratingCount”:[integer],  “originalGroupInfo”: {  “id”:[integer],  “key”:[string],  “type”:[GroupType],  “title”:[string],  “description”:[string],  “pictureUrl”:[string],  “avatarUrl”:[string],  “usersCount”:[integer],  “meGroupInfo”: {  “userGroupId”:[integer],  “status”:[GroupMemberStatus]  },  “levelToEditor”:[integer],  “visibilityStatus”:[ApiVisibilityStatus]  },  “groupInfo”: {  “id”:[integer],  “key”:[string],  “type”:[ GroupType],  “title”:[string],  “description”:[string],  “pictureUrl”:[string],  “avatarUrl”:[string],  “usersCount”:[integer],  “meGroupInfo”: {  “userGroupId”:[integer],  “status”:[GroupMemberStatus]  },  “levelToEditor”:[integer],  “visibilityStatus”:[ApiVisibilityStatus]  },  “authorInfo”: {  “userId”:[integer],  “displayName”:[string],  “pictureUrl”:[string],  “city”:[string]  “level”:[integer]  },  “likesInfo”: {  “count”:[integer],  “status”:[LikeStatus]  },  “visibilityStatus”:[ApiVisibilityStatus],  “userDiscussionInfo”: {  “lastTime”:[long],  “newCommentsCount”:[integer],  “isBookmarked”:[boolean]  }  },  ...  ] |
| **Error Response** | **Code:** 40x  **Content:** {  “code”:[ErrorInfo],  “message”:[string],  “developerMessage”:[string]  }  public enum PostType {  UNKNOWN,  USUAL,  INNER\_LINK,  OUTER\_LINK  }  public enum GroupType {  UNKNOWN,  CITY,  COMPANY,  COLLEGE,  TECH,  SPECIAL  }  public enum PostOrder {  UNKNOWN,  NEW,  OLD,  POPULAR  }  public enum PostFilter {  UNKNOWN,  ALL,  MY\_GROUPS  } |
| **Permissions** | All |

### 

### 

### **Get posts count**

|  |  |
| --- | --- |
| **URL** | /posts/count |
| **Method** | GET |
| **URL Params** | **Optional:**  query=String  type=PostType  groupKid=string  groupType=GroupType  ownerId=number  filter=PostFilter |
| **Data Params** | None |
| **Success Response** | **Code:** 200 OK  **Content:**[integer] |
| **Error Response** | **Code:** 40x  **Content:** {  “code”:[ErrorInfo],  “message”:[string],  “developerMessage”:[string]  } |
| **Notes** |  |
| **Permissions** | All |

### 

### 

### **Get post**

|  |  |
| --- | --- |
| **URL** | /posts/post{kid} |
| **Method** | GET |
| **URL Params** | **Required:**  kid=string |
| **Data Params** | None |
| **Success Response** | **Code:** 200 OK  **Content:** {  “id”:[integer],  “key”:[integer],  “type”:[PostType],  “title”:[string],  “description”:[string],  “pictureUrl”:[string],  “content”:[string],  “views”:[integer],  “commentsCount”:[integer],  “createdTime”:[Long],  “updatedTime”:[Long],  “rating”:[Double],  “ratingCount”:[integer],  “originalGroupInfo”: {  “id”:[integer],  “key”:[string],  “type”:[GroupType],  “title”:[string],  “description”:[string],  “pictureUrl”:[string],  “avatarUrl”:[string],  “usersCount”:[integer],  “meGroupInfo”: {  “userGroupId”:[integer],  “status”:[GroupMemberStatus]  },  “levelToEditor”:[integer],  “visibilityStatus”:[ApiVisibilityStatus]  },  “groupInfo”: null,  “authorInfo”: {  “userId”:[integer],  “displayName”:[string],  “pictureUrl”:[string],  “city”:[string]  “level”:[integer]  },  “likesInfo”: {  “count”:[integer],  “status”:[LikeStatus]  },  “visibilityStatus”:[ApiVisibilityStatus],  “userDiscussionInfo”: {  “lastTime”:[long],  “newCommentsCount”:[integer],  “isBookmarked”:[boolean]  }  } |
| **Error Response** | **Content:** {  “code”:POST\_NOT\_FOUND,  “message”:[string],  “developerMessage”:[string]  }  **Content:** {  “code”:INVALID\_ARGUMENT,  “message”:[string],  “developerMessage”:[string]  } |
| **Notes** | None |
| **Permissions** | All |

### 

### 