**Comprehensive Placeholder Test Cases for TemplateResolver**

Below is an extensive list of test cases covering edge cases , error scenarios , nested templates , context behavior , and special characters to ensure robust template resolution.

**1. Basic Placeholders**

| **Test Case** | **Example Template** | **Expected Behavior** | **Notes** |
| --- | --- | --- | --- |
| 1.1 | **${query\_id:1}** | Resolve query 1 result | Basic query resolution |
| 1.2 | **${user\_id}** | Replace with context variable**user\_id** | Basic variable replacement |
| 1.3 | **${query\_id:2[0].name}** | Resolve first item's name from query 2 | Query + array index + property |

**2. Array Index Variations**

| **Test Case** | **Example Template** | **Expected Behavior** | **Notes** |
| --- | --- | --- | --- |
| 2.1 | **${query\_id:3[0]}** | Resolve first item of query 3 | Static index |
| 2.2 | **${query\_id:3[${index}]}** | Resolve query 3 using**index**variable | Variable as index |
| 2.3 | **${query\_id:3[${query\_id:4}]}** | Use query 4 result as index for query 3 | Nested query as index |
| 2.4 | **${query\_id:3[invalid]}** | Throw error:**Invalid array index** | Non-numeric index |
| 2.5 | **${query\_id:3[${missingVar}]}** | Throw error:**variableNotFound** | Missing variable |

**3. Property Access**

| **Test Case** | **Example Template** | **Expected Behavior** | **Notes** |
| --- | --- | --- | --- |
| 3.1 | **${query\_id:5.name}** | Resolve**name**property of query 5 result | Basic property access |
| 3.2 | **${query\_id:5.data.user.name}** | Resolve nested property | Deep property access |
| 3.3 | **${query\_id:5["user-name"]}** | Resolve property with special char | Bracket notation |
| 3.4 | **${query\_id:5.missingProp}** | Return**undefined** | Missing property |

**4. Parameters in Placeholders**

| **Test Case** | **Example Template** | **Expected Behavior** | **Notes** |
| --- | --- | --- | --- |
| 4.1 | **${query\_id:6:{"filter":"active"}}** | Run query 6 with parameters | Static params |
| 4.2 | **${query\_id:6:{"id":${user\_id}}}** | Use**user\_id**variable in params | Dynamic params |
| 4.3 | **${query\_id:6:{"id":${query\_id:7}}}** | Use query 7 result in params | Nested query in params |
| 4.4 | **${query\_id:6:invalidJSON}** | Throw error:**parseParams:catch** | Invalid JSON |

**5. Error Scenarios**

| **Test Case** | **Example Template** | **Expected Behavior** | **Notes** |
| --- | --- | --- | --- |
| 5.1 | **${missingQuery:99}** | Throw error:**Query not found** | Invalid query ID |
| 5.2 | **${missingVar}** | Throw warning:**variableNotFound** | Missing variable |
| 5.3 | **${query\_id:10[1000]}** | Return**undefined** | Out-of-bounds index |
| 5.4 | **${query\_id:11:{"param":${missingVar}}}** | Throw error:**variableNotFound** | Missing var in params |
| 5.5 | **${query\_id:12[${circular}]}**where**circular=${query\_id:12[${circular}]}** | Throw error:**maxDepth** | Circular reference |

**6. Context Behavior**

| **Test Case** | **Example Template** | **Expected Behavior** | **Notes** |
| --- | --- | --- | --- |
| 6.1 | **${parentVar}**in ChildContext | Resolve from parent | Variable inheritance |
| 6.2 | **${childVar}**in ChildContext | Resolve from child | Override parent variable |
| 6.3 | **${query\_id:13:{"id":1}}** | Use cached result | Result caching |
| 6.4 | **${query\_id:13:{"id":2}}** | Run query again | Different parameters → new result |

**7. Recursion & Depth Limits**

| **Test Case** | **Example Template** | **Expected Behavior** | **Notes** |
| --- | --- | --- | --- |
| 7.1 | **${query\_id:14[${query\_id:15}]}** | Resolve query 15 → use as index for 14 | Single-level recursion |
| 7.2 | **${query\_id:16[${query\_id:17[${query\_id:18}]}]}** | Resolve 18 → 17 → 16 | Multi-level recursion |
| 7.3 | **${query\_id:19[${query\_id:19}]}** | Throw error:**maxDepth** | Circular recursion |
| 7.4 | **${deepNested}**where**deepNested=${a},${b},${c},${d},...** | Resolve recursively | Test max depth limit |

**8. Special Characters**

| **Test Case** | **Example Template** | **Expected Behavior** | **Notes** |
| --- | --- | --- | --- |
| 8.1 | **${query\_id:20["key.with.dots"]}** | Resolve property with dots | Escaped dots |
| 8.2 | **${query\_id:21:{"filter":"va{l}ue"}** | Pass literal**{}**in params | Literal braces |
| 8.3 | **${"escaped"}** | Return**${"escaped"}** | Escaped quotes |
| 8.4 | **${query\_id:22["sp ace"]}** | Resolve property with space | Space in key |
| 8.5 | **${query\_id:23:{"key":"\"quoted\""}}** | Pass quoted value in params | Quoted string in JSON |

**9. Caching & Parameterization**

| **Test Case** | **Example Template** | **Expected Behavior** | **Notes** |
| --- | --- | --- | --- |
| 9.1 | **${query\_id:24}**(same params) | Use cached result | No re-run |
| 9.2 | **${query\_id:24:{"id":1}}** | Cache result with params | Parameterized caching |
| 9.3 | **${query\_id:24:{"id":2}}** | Run again with new params | New cache entry |
| 9.4 | **${query\_id:25:{"a":1,"b":2}}** | Sort keys →**{"a":1,"b":2}** | Stable param key |

**10. Complex & Edge Cases**

| **Test Case** | **Example Template** | **Expected Behavior** | **Notes** |
| --- | --- | --- | --- |
| 10.1 | **""**(empty string) | Return**""** | Empty input |
| 10.2 | **${query\_id:26}${query\_id:27}** | Resolve both queries | Multiple placeholders |
| 10.3 | **Text before ${query\_id:28} text after** | Replace only placeholder | Mixed text |
| 10.4 | **[${query\_id:29}]** | Resolve query 29 in array | Array context |
| 10.5 | **${query\_id:30:{"arr":[${query\_id:31}]}}** | Use query 31 in array | Nested array in params |

**11. Context-Specific Tests**

| **Test Case** | **Example Template** | **Expected Behavior** | **Notes** |
| --- | --- | --- | --- |
| 11.1 | **${query\_id:32:{"user":${user\_id}}}** | Use**user\_id**from context | Variable in params |
| 11.2 | **${query\_id:33[${phoneapi}].id}**where**phoneapi=0** | Resolve to**id**at index 0 | Real-world example |
| 11.3 | **${query\_id:34:{"filter":${missingFilter}}}** | Throw error:**variableNotFound** | Missing var in params |
| 11.4 | **${query\_id:35:{"id":${query\_id:36}}}** | Resolve query 36 → use in query 35 | Query result in params |

**Summary Table**

| **Category** | **Test Count** | **Description** |
| --- | --- | --- |
| Basic | 3 | Core placeholder functionality |
| Array Index | 5 | Index variations |
| Property Access | 4 | Object property handling |
| Parameters | 4 | Param injection |
| Errors | 5 | Failure scenarios |
| Context | 4 | Parent/child behavior |
| Recursion | 4 | Nested templates |
| Special Chars | 5 | Escaped/complex keys |
| Caching | 4 | Result reuse |
| Complex Cases | 5 | Mixed inputs |
| Context-Specific | 4 | Real-world scenarios |
| Total | 43+ | Comprehensive coverage |

These test cases ensure robust template resolution , error handling , and correct context behavior for the TemplateResolver system.