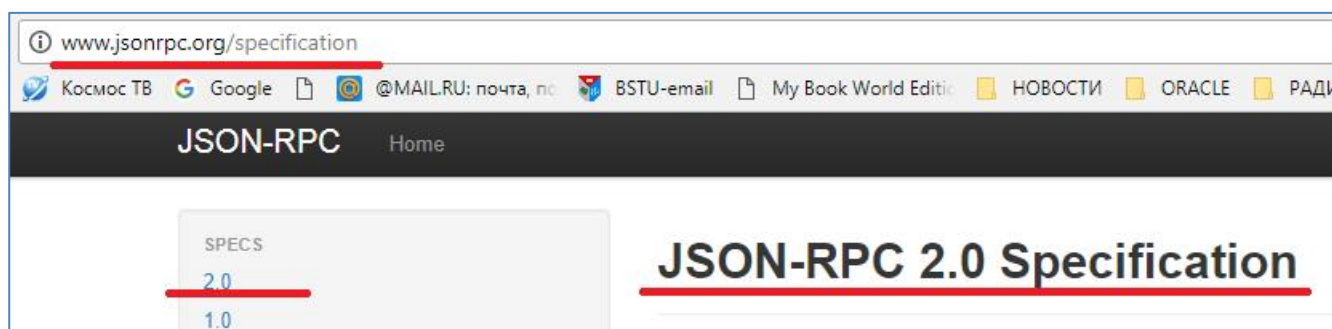


JSON-RPC

1. **JSON-RPC**: протокол удаленного вызова процедур, использующий формат JSON для передачи сообщений.
2. **JSON-RPC**: последняя версия 2.0.
3. **JSON-RPC**: поддержка



4. **JSON-RPC**: транспорт HTTP/HTTPS или TCP Socket.
5. **JSON-RPC**: JSON RFC 4627



6. **JSON-RPC**: объект запроса

4 Request object

A rpc call is represented by sending a Request object to a Server. The Request object has the following members:

jsonrpc
A String specifying the version of the JSON-RPC protocol. MUST be exactly "2.0".

method
A String containing the name of the method to be invoked. Method names that begin with the word rpc followed by a period character (U+002E or ASCII 46) are reserved for rpc-internal methods and extensions and MUST NOT be used for anything else.

params
A Structured value that holds the parameter values to be used during the invocation of the method. This member MAY be omitted.

id
An identifier established by the Client that MUST contain a String, Number, or NULL value if included. If it is not included it is assumed to be a notification. The value SHOULD normally not be Null [1] and Numbers SHOULD NOT contain fractional parts [2]

7. JSON-RPC: обект отбета

5 Response object

When a rpc call is made, the Server MUST reply with a Response, except for in the case of Notifications. The Response is expressed as a single JSON Object, with the following members:

jsonrpc

A String specifying the version of the JSON-RPC protocol. MUST be exactly "2.0".

result

This member is REQUIRED on success.

This member MUST NOT exist if there was an error invoking the method.

The value of this member is determined by the method invoked on the Server.

error

This member is REQUIRED on error.

This member MUST NOT exist if there was no error triggered during invocation.

The value for this member MUST be an Object as defined in section 5.1.

id

This member is REQUIRED.

It MUST be the same as the value of the id member in the Request Object.

If there was an error in detecting the id in the Request object (e.g. Parse error/Invalid Request), it MUST be Null.

Either the result member or error member MUST be included, but both members MUST NOT be included.

8. JSON-RPC: обект отбета (ошибка)

5.1 Error object

When a rpc call encounters an error, the Response Object MUST contain the error member with a value that is a Object with the following members:

code

A Number that indicates the error type that occurred.

This MUST be an integer.

message

A String providing a short description of the error.

The message SHOULD be limited to a concise single sentence.

data

A Primitive or Structured value that contains additional information about the error.

This may be omitted.

The value of this member is defined by the Server (e.g. detailed error information, nested errors etc.).

The error codes from and including -32768 to -32000 are reserved for pre-defined errors. Any code within this range, but not defined explicitly below is reserved for future use. The error codes are nearly the same as those suggested for XML-RPC at the following url: http://xmlrpc-epi.sourceforge.net/specs/rfc.fault_codes.php

code	message	meaning
-32700	Parse error	Invalid JSON was received by the server. An error occurred on the server while parsing the JSON text.
-32600	Invalid Request	The JSON sent is not a valid Request object.
-32601	Method not found	The method does not exist / is not available.
-32602	Invalid params	Invalid method parameter(s).
-32603	Internal error	Internal JSON-RPC error.
-32000 to -32099	Server error	Reserved for implementation-defined server-errors.

The remainder of the space is available for application defined errors.

9. **JSON-RPC**: пример rpc с позиционными параметрами

```
--> {"jsonrpc": "2.0", "method": "subtract", "params": [42, 23], "id": 1}
<-- {"jsonrpc": "2.0", "result": 19, "id": 1}
```

```
--> {"jsonrpc": "2.0", "method": "subtract", "params": [23, 42], "id": 2}
<-- {"jsonrpc": "2.0", "result": -19, "id": 2}
```

10. **JSON-RPC**: пример rpc с именованными параметрами

```
--> {"jsonrpc": "2.0", "method": "subtract", "params": {"subtrahend": 23, "minuend": 42}, "id": 3}
<-- {"jsonrpc": "2.0", "result": 19, "id": 3}
```

```
--> {"jsonrpc": "2.0", "method": "subtract", "params": {"minuend": 42, "subtrahend": 23}, "id": 4}
<-- {"jsonrpc": "2.0", "result": 19, "id": 4}
```

11. **JSON-RPC**: пример rpc с уведомлением (запрос без ответа)

```
--> {"jsonrpc": "2.0", "method": "update", "params": [1,2,3,4,5]}
```

```
--> {"jsonrpc": "2.0", "method": "foobar"}
```

12. **JSON-RPC**: примеры rpc с ответами-ошибками

```
--> {"jsonrpc": "2.0", "method": "foobar", "id": "1"}
<-- {"jsonrpc": "2.0", "error": {"code": -32601, "message": "Method not found"}, "id": "1"}
```

```
--> {"jsonrpc": "2.0", "method": "foobar", "id": "1"}
<-- {"jsonrpc": "2.0", "error": {"code": -32601, "message": "Method not found"}, "id": "1"}
```

```
--> {"jsonrpc": "2.0", "method": 1, "params": "bar"}
<-- {"jsonrpc": "2.0", "error": {"code": -32600, "message": "Invalid Request"}, "id": null}
```

13. **JSON-RPC**: примеры rpc с пакетом запросов

```
--> [
  {"jsonrpc": "2.0", "method": "sum", "params": [1,2,4], "id": "1"},
  {"jsonrpc": "2.0", "method": "notify_hello", "params": [7]},
  {"jsonrpc": "2.0", "method": "subtract", "params": [42,23], "id": "2"},
  {"foo": "boo"},
  {"jsonrpc": "2.0", "method": "foo.get", "params": {"name": "myself"}, "id": "5"},
  {"jsonrpc": "2.0", "method": "get_data", "id": "9"}
]
<-- [
  {"jsonrpc": "2.0", "result": 7, "id": "1"},
  {"jsonrpc": "2.0", "result": 19, "id": "2"},
  {"jsonrpc": "2.0", "error": {"code": -32600, "message": "Invalid Request"}, "id": null},
  {"jsonrpc": "2.0", "error": {"code": -32601, "message": "Method not found"}, "id": "5"},
  {"jsonrpc": "2.0", "result": ["hello", 5], "id": "9"}
]
```

```
--> [1]
<-- [
  {"jsonrpc": "2.0", "error": {"code": -32600, "message": "Invalid Request"}, "id": null}
]
```

```
--> [
  {"jsonrpc": "2.0", "method": "notify_sum", "params": [1,2,4]},
  {"jsonrpc": "2.0", "method": "notify_hello", "params": [7]}
]
<-- //Nothing is returned for all notification batches
```

```
--> [
  {"jsonrpc": "2.0", "method": "sum", "params": [1,2,4], "id": "1"},
  {"jsonrpc": "2.0", "method":
}
<-- {"jsonrpc": "2.0", "error": {"code": -32700, "message": "Parse error"}, "id": null}
```

14. JSON-RPC: начало примера

```
namespace JsonRPCServer.Controllers
{
    public class CalcController : ApiController
    {
        [HttpPost]
        public ResJsonRPC Post([FromBody] ReqJsonRPC jsonreq)
        {
            int sum = 0;
            for(int i = 0; i < jsonreq.Params.Length; i++) sum += jsonreq.Params[i];

            ResJsonRPC rc = new ResJsonRPC()
            {
                Id = jsonreq.Id,
                Jsonrpc = jsonreq.Jsonrpc,
                Method = jsonreq.Method,
                Result = sum
            };
            return rc;
        }

        public class ReqJsonRPC
        {
            public string Id { get; set; }
            public string Jsonrpc { get; set; }
            public string Method { get; set; }
            public int [] Params { get; set; }
        }

        public class ResJsonRPC
        {
            public string Id { get; set; }
            public string Jsonrpc { get; set; }
            public string Method { get; set; }
            public int Result { get; set; }
        }
    }
}
```


POST ▼ http://localhost:59620/jsonrpc/Calc	
Authorization Headers (2) Body ● Pre-request Script Tests	
Key	Value
<input checked="" type="checkbox"/> Content-Length	56
<input checked="" type="checkbox"/> Content-Type	application/json

POST ▼ http://localhost:59620/jsonrpc/Calc	
Authorization Headers (2) Body ● Pre-request Script Tests	
<input type="radio"/> form-data <input type="radio"/> x-www-form-urlencoded <input checked="" type="radio"/> raw <input type="radio"/> binary JSON (application/json) ▼	
1	<code>{"jsonrpc": "2.0", "method": "Add", "id": "1", "params": [1, 4]}</code>

Body Cookies Headers (10) Test Results	
Pretty Raw Preview JSON ▼	
1	<code>{</code>
2	<code>"Id": "1",</code>
3	<code>"Jsonrpc": "2.0",</code>
4	<code>"Method": "Add",</code>
5	<code>"Result": 5</code>
6	<code>}</code>

15. JSON-RPC:
16. JSON-RPC:
17. JSON-RPC:
18. JSON-RPC:
19. JSON-RPC:
20. JSON-RPC:
21. JSON-RPC:
22. JSON-RPC:
23. JSON-RPC:
24. JSON-RPC:

- 25. **JSON-RPC:**
- 26. **JSON-RPC:**
- 27. **JSON-RPC:**
- 28. **JSON-RPC:**
- 29. **JSON-RPC:**
- 30. **JSON-RPC:**
- 31. **JSON-RPC:**
- 32. **JSON-RPC:**
- 33. **JSON-RPC:**
- 34. **JSON-RPC:**
- 35.
- 36.
- 37.