



MEETUP ONLINE '21

# НОВЫЕ ВЕБХУКИ: КАК НАПИСАТЬ ХОРОШУЮ ИНТЕГРАЦИЮ

ТИХОН УСКОВ  
ИНЖЕНЕР-ИНТЕГРАТОР

ZABBIX



MEETUP ONLINE '21

Новые вебхуки: как написать хорошую интеграцию

# WEBHOOK – ЧТО ЗА ЗВЕРЬ?



MEETUP ONLINE '21

Новые вебхуки: как написать хорошую интеграцию

# НОВЫЕ ИНТЕГРАЦИИ

# НОВЫЕ ИНТЕГРАЦИИ



brevis.one



- ✓ iTop – система управления ИТ-услугами
- ✓ VictorOps (Splunk On-Call) – платформа для управления инцидентами
- ✓ Brevis.one – СМС-шлюзы
- ✓ Rocket.Chat – чат-платформа с открытым исходным кодом



MEETUP ONLINE '21

Новые вебхуки: как написать хорошую интеграцию

# РАЗБОР КОДА ИНТЕГРАЦИЙ

# СТРУКТУРА

- ✓ Функциональный класс
- ✓ Логика:
  - ✓ Объявление переменных
  - ✓ Валидация параметров
  - ✓ Запросы к API
  - ✓ Обработка и отправка результата
- ✓ Обработка ошибок

```
setParams: function (params) {
  if (typeof params !== 'object') {
    return;
  }
  RocketChat.params = params;
  if (RocketChat.params.url && RocketChat.params.api_url) {
    if (!RocketChat.params.url.endsWith('/')) {
      RocketChat.params.url += '/';
    }
    if (!RocketChat.params.api_url.endsWith('/')) {
      RocketChat.params.api_url += '/';
    }
    if (RocketChat.params.api_url.startsWith('/')) {
      RocketChat.params.api_url = RocketChat.params.api_url.substring(1);
    }
    RocketChat.params.url += RocketChat.params.api_url;
  },
  setProxy: function (HTTPProxy) {
    RocketChat.HTTPProxy = HTTPProxy;
  },
  addFields: function (fields) {
    var data = [];
    if (typeof fields === 'object' && Object.keys(fields).length) {
      Object.keys(fields)
        .forEach(function(field) {
          if (fields[field] === '') {
            Zabbix.log(4, '[ RocketChat Webhook ] Field "' + field +
              '" can\'t be empty. The field ignored.');
```

# WORKFLOW

```
// Создание класса.
var SomeClass = {
  params: {},

  // Описание методов.
  someMethod: function (value) {
    SomeClass.params.someParam = value;
  };
};

try {
  var params = JSON.parse(value),
      result = {tags: {}};

  // Тут какая-то логика.

  return JSON.stringify(result);
}
// Обработка ошибок.
catch (error) {
  Zabbix.log(3, '[ SomeHook Webhook ] ERROR: ' + error);
  throw 'Sending failed: ' + error;
}
```

# WORKFLOW: ПЕРЕМЕННЫЕ И МАССИВЫ

```
var params = JSON.parse(value),  
    response,  
    result = {tags: {}},  
    fields = {},  
    rc = {},  
    update = {},  
    required_params = ['alert_subject', 'alert_message'];
```



# WORKFLOW: ПЕРЕМЕННЫЕ И МАССИВЫ

```
var params = JSON.parse(value),
    response,
    result = {tags: {}},
    fields = {},
    rc = {},
    update = {},
    required_params = ['alert_subject', 'alert_message'];
```

```
Object.keys(params)
  .forEach(function (key) {
    if (key.startsWith('rc_')) {
      rc[key.substring(3)] = params[key];
    }
    else if (key.startsWith('field_')) {
      fields[key.substring(6)] = params[key];
    }
    else if (key.startsWith('event_update_')) {
      update[key.substring(13)] = params[key];
    }
    else if (required_params.indexOf(key) !== -1 && params[key] === '') {
      throw 'Parameter "' + key + '" can\'t be empty.';
    }
  });
```

field_1_full:Host	{HOST.NAME} [{HOST.IP}]	<a href="#">Remove</a>
field_2_short:Severity	{EVENT.SEVERITY}	<a href="#">Remove</a>
field_5_short:Operational data	{EVENT.OPDATA}	<a href="#">Remove</a>
field_999_full_p:Trigger description	{TRIGGER.DESCRPTION}	<a href="#">Remove</a>
rc_api_url	api/v1/	<a href="#">Remove</a>
rc_msg_id	{EVENT.TAGS.__zbx_rc_id}	<a href="#">Remove</a>
rc_room_id	{EVENT.TAGS.__zbx_rc rid}	<a href="#">Remove</a>
rc_send_to	{ALERT.SENDTO}	<a href="#">Remove</a>
rc_title_link	{\$ZABBIX.URL}/tr_events.php?trig	<a href="#">Remove</a>
rc_url	<PLACE YOUR INSTANCE URL H	<a href="#">Remove</a>
rc_user_id	<PLACE USER ID HERE>	<a href="#">Remove</a>
rc_user_token	<PLACE TOKEN HERE>	<a href="#">Remove</a>

# WORKFLOW: ПЕРЕМЕННЫЕ И МАССИВЫ

```
var params = JSON.parse(value),
    response,
    result = {tags: {}},
    fields = {},
    rc = {},
    update = {},
    required_params = ['alert_subject', 'alert_message'];
```

```
Object.keys(params)
  .forEach(function (key) {
    if (key.startsWith('rc_')) {
      rc[key.substring(3)] = params[key];
    }
    else if (key.startsWith('field_')) {
      fields[key.substring(6)] = params[key];
    }
    else if (key.startsWith('event_update_')) {
      update[key.substring(13)] = params[key];
    }
    else if (required_params.indexOf(key) !== -1 && params[key] === '') {
      throw 'Parameter "' + key + '" can\'t be empty.';
    }
  });
```

field_1_full:Host	{HOST.NAME} [{HOST.IP}]	<a href="#">Remove</a>
field_2_short:Severity	{EVENT.SEVERITY}	<a href="#">Remove</a>
field_5_short:Operational data	{EVENT.OPDATA}	<a href="#">Remove</a>
field_999_full_p:Trigger description	{TRIGGER.DESCRPTION}	<a href="#">Remove</a>
rc_api_url	api/v1/	<a href="#">Remove</a>
rc_msg_id	{EVENT.TAGS.__zbx_rc_id}	<a href="#">Remove</a>
rc_room_id	{EVENT.TAGS.__zbx_rc rid}	<a href="#">Remove</a>
rc_send_to	{ALERT.SENDTO}	<a href="#">Remove</a>
rc_title_link	{\$ZABBIX.URL}/tr_events.php?trig	<a href="#">Remove</a>
rc_url	<PLACE YOUR INSTANCE URL H	<a href="#">Remove</a>
rc_user_id	<PLACE USER ID HERE>	<a href="#">Remove</a>
rc_user_token	<PLACE TOKEN HERE>	<a href="#">Remove</a>

# WORKFLOW: ПЕРЕМЕННЫЕ И МАССИВЫ

```
var params = JSON.parse(value),
    response,
    result = {tags: {}},
    fields = {},
    rc = {},
    update = {},
    required_params = ['alert_subject', 'alert_message'];
```

```
Object.keys(params)
  .forEach(function (key) {
    if (key.startsWith('rc_')) {
      rc[key.substring(3)] = params[key];
    }
    else if (key.startsWith('field_')) {
      fields[key.substring(6)] = params[key];
    }
    else if (key.startsWith('event_update_')) {
      update[key.substring(13)] = params[key];
    }
    else if (required_params.indexOf(key) !== -1 && params[key] === '') {
      throw 'Parameter "' + key + '" can\'t be empty.';
    }
  });
```

field_1_full:Host	{HOST.NAME} [{HOST.IP}]	<a href="#">Remove</a>
field_2_short:Severity	{EVENT.SEVERITY}	<a href="#">Remove</a>
field_5_short:Operational data	{EVENT.OPDATA}	<a href="#">Remove</a>
field_999_full_p:Trigger description	{TRIGGER.DESCRPTION}	<a href="#">Remove</a>
rc_api_url	api/v1/	<a href="#">Remove</a>
rc_msg_id	{EVENT.TAGS.__zbx_rc_id}	<a href="#">Remove</a>
rc_room_id	{EVENT.TAGS.__zbx_rc rid}	<a href="#">Remove</a>
rc_send_to	{ALERT.SENDTO}	<a href="#">Remove</a>
rc_title_link	{\$ZABBIX.URL}/tr_events.php?trig	<a href="#">Remove</a>
rc_url	<PLACE YOUR INSTANCE URL H	<a href="#">Remove</a>
rc_user_id	<PLACE USER ID HERE>	<a href="#">Remove</a>
rc_user_token	<PLACE TOKEN HERE>	<a href="#">Remove</a>

# WORKFLOW: ВАЛИДАЦИЯ ПАРАМЕТРОВ

```
if ([0, 1, 2, 3].indexOf(parseInt(params.event_source)) === -1) {  
    throw 'Incorrect "event_source" parameter given: ' + params.event_source + '\nMust be 0-3.';  
}
```

alert_message	{ALERT.MESSAGE}	<a href="#">Remove</a>
alert_subject	{ALERT.SUBJECT}	<a href="#">Remove</a>
event_nseverity	{EVENT.NSEVERITY}	<a href="#">Remove</a>
event_source	{EVENT.SOURCE}	<a href="#">Remove</a>
event_update_action	{EVENT.UPDATE.ACTION}	<a href="#">Remove</a>
event_update_message	{EVENT.UPDATE.MESSAGE}	<a href="#">Remove</a>
event_update_status	{EVENT.UPDATE.STATUS}	<a href="#">Remove</a>
event_update_user	{USER.FULLNAME}	<a href="#">Remove</a>
event_value	{EVENT.VALUE}	<a href="#">Remove</a>
use_default_message	false	<a href="#">Remove</a>

# WORKFLOW: ВАЛИДАЦИЯ ПАРАМЕТРОВ

```
if ([0, 1, 2, 3].indexOf(parseInt(params.event_source)) === -1) {  
    throw 'Incorrect "event_source" parameter given: ' + params.event_source + '\nMust be 0-3.';  
}  
  
// Принудительно устанавливаем параметры для не триггерных событий.  
if (params.event_source !== '0') {  
    params.use_default_message = 'true';  
    params.event_nseverity = '0';  
    params.rc_title_link = false;  
}
```

alert_message	{ALERT.MESSAGE}	<a href="#">Remove</a>
alert_subject	{ALERT.SUBJECT}	<a href="#">Remove</a>
event_nseverity	{EVENT.NSEVERITY}	<a href="#">Remove</a>
event_source	{EVENT.SOURCE}	<a href="#">Remove</a>
event_update_action	{EVENT.UPDATE.ACTION}	<a href="#">Remove</a>
event_update_message	{EVENT.UPDATE.MESSAGE}	<a href="#">Remove</a>
event_update_status	{EVENT.UPDATE.STATUS}	<a href="#">Remove</a>
event_update_user	{USER.FULLNAME}	<a href="#">Remove</a>
event_value	{EVENT.VALUE}	<a href="#">Remove</a>
use_default_message	false	<a href="#">Remove</a>

# WORKFLOW: ВАЛИДАЦИЯ ПАРАМЕТРОВ

```
if ([0, 1, 2, 3].indexOf(parseInt(params.event_source)) === -1) {  
    throw 'Incorrect "event_source" parameter given: ' + params.event_source + '\nMust be 0-3.';  
}  
  
// Принудительно устанавливаем параметры для не триггерных событий.  
if (params.event_source !== '0') {  
    params.use_default_message = 'true';  
    params.event_nseverity = '0';  
    params.rc_title_link = false;  
}
```

alert_message	{ALERT.MESSAGE}	<a href="#">Remove</a>
alert_subject	{ALERT.SUBJECT}	<a href="#">Remove</a>
event_nseverity	{EVENT.NSEVERITY}	<a href="#">Remove</a>
event_source	{EVENT.SOURCE}	<a href="#">Remove</a>
event_update_action	{EVENT.UPDATE.ACTION}	<a href="#">Remove</a>
event_update_message	{EVENT.UPDATE.MESSAGE}	<a href="#">Remove</a>
event_update_status	{EVENT.UPDATE.STATUS}	<a href="#">Remove</a>
event_update_user	{USER.FULLNAME}	<a href="#">Remove</a>
event_value	{EVENT.VALUE}	<a href="#">Remove</a>
use_default_message	false	<a href="#">Remove</a>

# WORKFLOW: ЛОГИКА

```
// Отправляем {ALERT.MESSAGE}, если use_default_message === true.
if (params.use_default_message.toLowerCase() === 'true') {

    response = RocketChat.postMessage(true, params.alert_message);

    result.tags.__zbx_rc_id = response.id;

}
// Отправляем нотификацию для триггерных событий.
else if (params.event_value === '1' && update.status === '0') {

    response = RocketChat.postMessage(false, params.alert_message, fields);

    result.tags.__zbx_rc_id = response.id;

}
// Отправляем сообщение в существующий тред для триггерных событий (UPDATE и RESOLVE).
else {
    RocketChat.sendMessage(update, fields);
}

return JSON.stringify(result);
```

Process tags	<input checked="" type="checkbox"/>
Include event menu entry	<input checked="" type="checkbox"/>
* Menu entry name	<input type="text" value="Rocket.Chat"/>
* Menu entry URL	<input type="text" value="{EVENT.TAGS.__zbx_rc_msg_url}"/>

# WORKFLOW: ЛОГИКА

```
// Отправляем {ALERT.MESSAGE}, если use_default_message === true.
if (params.use_default_message.toLowerCase() === 'true') {

    response = RocketChat.postMessage(true, params.alert_message);

    result.tags.__zbx_rc_id = response.id;

}
// Отправляем нотификацию для триггерных событий.
else if (params.event_value === '1' && update.status === '0') {

    response = RocketChat.postMessage(false, params.alert_message, fields);

    result.tags.__zbx_rc_id = response.id;

}
// Отправляем сообщение в существующий тред для триггерных событий (UPDATE и RESOLVE).
else {
    RocketChat.sendMessage(update, fields);
}

return JSON.stringify(result);
```

Process tags	<input checked="" type="checkbox"/>
Include event menu entry	<input checked="" type="checkbox"/>
* Menu entry name	<input type="text" value="Rocket.Chat"/>
* Menu entry URL	<input type="text" value="{EVENT.TAGS.__zbx_rc_msg_url}"/>



# WORKFLOW: ЛОГИКА

```
// Отправляем {ALERT.MESSAGE}, если use_default_message === true.  
if (params.use_default_message.toLowerCase() === 'true') {  
    response = RocketChat.postMessage(true, params.alert_message);  
    result.tags.__zbx_rc_id = response.id;  
}  
// Отправляем нотификацию для триггерных событий.  
else if (params.event_value === '1' && update.status === '0') {  
    response = RocketChat.postMessage(false, params.alert_message, fields);  
    result.tags.__zbx_rc_id = response.id;  
}  
// Отправляем сообщение в существующий тред для триггерных событий (UPDATE и RESOLVE).  
else {  
    RocketChat.sendMessage(update, fields);  
}  
  
return JSON.stringify(result);
```

Process tags	<input checked="" type="checkbox"/>
Include event menu entry	<input checked="" type="checkbox"/>
* Menu entry name	<input type="text" value="Rocket.Chat"/>
* Menu entry URL	<input type="text" value="{EVENT.TAGS.__zbx_rc_msg_url}"/>

# WORKFLOW: ЛОГИКА

```
// Отправляем {ALERT.MESSAGE}, если use_default_message === true.  
if (params.use_default_message.toLowerCase() === 'true') {  
    response = RocketChat.postMessage(true, params.alert_message);  
    result.tags.__zbx_rc_id = response.id;  
}  
// Отправляем нотификацию для триггерных событий.  
else if (params.event_value === '1' && update.status === '0') {  
    response = RocketChat.postMessage(false, params.alert_message, fields);  
    result.tags.__zbx_rc_id = response.id;  
}  
// Отправляем сообщение в существующий тред для триггерных событий (UPDATE и RESOLVE).  
else {  
    RocketChat.sendMessage(update, fields);  
}  
return JSON.stringify(result);
```

Process tags	<input checked="" type="checkbox"/>
Include event menu entry	<input checked="" type="checkbox"/>
* Menu entry name	<input type="text" value="Rocket.Chat"/>
* Menu entry URL	<input type="text" value="{EVENT.TAGS.__zbx_rc_msg_url}"/>

# WORKFLOW: ЛОГИКА

```
// Отправляем {ALERT.MESSAGE}, если use_default_message === true.
if (params.use_default_message.toLowerCase() === 'true') {

    response = RocketChat.postMessage(true, params.alert_message);

    result.tags.__zbx_rc_id = response.id;

}
// Отправляем нотификацию для триггерных событий.
else if (params.event_value === '1' && update.status === '0') {

    response = RocketChat.postMessage(false, params.alert_message, fields);

    result.tags.__zbx_rc_id = response.id;

}
// Отправляем сообщение в существующий тред для триггерных событий (UPDATE и RESOLVE).
else {
    RocketChat.sendMessage(update, fields);
}

return JSON.stringify(result);
```

Process tags	<input checked="" type="checkbox"/>
Include event menu entry	<input checked="" type="checkbox"/>
* Menu entry name	<input type="text" value="Rocket.Chat"/>
* Menu entry URL	<input type="text" value="{EVENT.TAGS.__zbx_rc_msg_url}"/>

# ОБЗОР ГЕНЕРАТОРА ПОЛЕЙ

```
addFields: function (fields) {
  var data = [];

  if (typeof fields === 'object' && Object.keys(fields).length) {
    Object.keys(fields)
      .forEach(function(field) {
        if (fields[field] === '') {
          Zabbix.log(4, '[ RocketChat Webhook ] Field "' + field +
            '" can\'t be empty. The field ignored.');
```

```
        }
        else {
          try {
            var parts = field.split(':'),
                prefix = parts[0].split('_');

            if (typeof prefix[2] === 'undefined'
                || (prefix[2] === 'p' && params.event_value === '1')
                || (prefix[2] === 'r' && params.event_value === '0')) {
              data.push({
                title: field.substring(field.indexOf(':') + 1),
                value: fields[field],
                short: prefix[1] === 'short'
              });
            }
          }
          catch (error) {
            Zabbix.log(4, '[ RocketChat Webhook ] Can\'t parse field "' + field +
              '". The field ignored.');
```

```
          }
        }
      });
  }

  return data;
}
```

field_1_full:Host	{HOST.NAME} {{HOST.IP}}	<a href="#">Remove</a>
field_2_short:Severity	{EVENT.SEVERITY}	<a href="#">Remove</a>
field_3_short:Event time	{EVENT.DATE} {EVENT.TIME}	<a href="#">Remove</a>
field_3_short_r:Recovery time	{EVENT.RECOVERY.DATE} {EVE	<a href="#">Remove</a>
field_4_short_r:Event duration	{EVENT.DURATION}	<a href="#">Remove</a>
field_5_short:Operational data	{EVENT.OPDATA}	<a href="#">Remove</a>
field_999_full_p:Trigger descriptio	{TRIGGER.DESCRPTION}	<a href="#">Remove</a>

# ОБЗОР ГЕНЕРАТОРА ПОЛЕЙ

```
addFields: function (fields) {
  var data = [];

  if (typeof fields === 'object' && Object.keys(fields).length) {
    Object.keys(fields)
      .forEach(function(field) {
        if (fields[field] === '') {
          Zabbix.log(4, '[ RocketChat Webhook ] Field "' + field +
            '" can\'t be empty. The field ignored.');
```

```
        }
        else {
          try {
            var parts = field.split(':'),
                prefix = parts[0].split('_');

            if (typeof prefix[2] === 'undefined'
                || (prefix[2] === 'p' && params.event_value === '1')
                || (prefix[2] === 'r' && params.event_value === '0')) {
              data.push({
                title: field.substring(field.indexOf(':') + 1),
                value: fields[field],
                short: prefix[1] === 'short'
              });
            }
          }
          catch (error) {
            Zabbix.log(4, '[ RocketChat Webhook ] Can\'t parse field "' + field +
              '". The field ignored.');
```

```
          }
        }
      });
  }

  return data;
}
```

field_1_full:Host	{HOST.NAME} {{HOST.IP}}	<a href="#">Remove</a>
field_2_short:Severity	{EVENT.SEVERITY}	<a href="#">Remove</a>
field_3_short:Event time	{EVENT.DATE} {EVENT.TIME}	<a href="#">Remove</a>
field_3_short_r:Recovery time	{EVENT.RECOVERY.DATE} {EVE	<a href="#">Remove</a>
field_4_short_r:Event duration	{EVENT.DURATION}	<a href="#">Remove</a>
field_5_short:Operational data	{EVENT.OPDATA}	<a href="#">Remove</a>
field_999_full_p:Trigger descriptio	{TRIGGER.DESCRPTION}	<a href="#">Remove</a>

# ОБЗОР ГЕНЕРАТОРА ПОЛЕЙ

```
addFields: function (fields) {
  var data = [];

  if (typeof fields === 'object' && Object.keys(fields).length) {
    Object.keys(fields)
      .forEach(function(field) {
        if (fields[field] === '') {
          Zabbix.log(4, '[ RocketChat Webhook ] Field "' + field +
            '" can\'t be empty. The field ignored.');
```

return data;

field_1_full:Host	{HOST.NAME} {{HOST.IP}}	<a href="#">Remove</a>
field_2_short:Severity	{EVENT.SEVERITY}	<a href="#">Remove</a>
field_3_short:Event time	{EVENT.DATE} {EVENT.TIME}	<a href="#">Remove</a>
field_3_short_r:Recovery time	{EVENT.RECOVERY.DATE} {EVE	<a href="#">Remove</a>
field_4_short_r:Event duration	{EVENT.DURATION}	<a href="#">Remove</a>
field_5_short:Operational data	{EVENT.OPDATA}	<a href="#">Remove</a>
field_999_full_p:Trigger descriptio	{TRIGGER.DESCRPTION}	<a href="#">Remove</a>

# ОБЗОР ГЕНЕРАТОРА ПОЛЕЙ

```
addFields: function (fields) {
  var data = [];

  if (typeof fields === 'object' && Object.keys(fields).length) {
    Object.keys(fields)
      .forEach(function(field) {
        if (fields[field] === '') {
          Zabbix.log(4, '[ RocketChat Webhook ] Field "' + field +
            '" can\'t be empty. The field ignored.');
```

Object.keys(fields)  
 .forEach(function(field) {

```
        }
        else {
          try {
            var parts = field.split(':'),
                prefix = parts[0].split('_');

            if (typeof prefix[2] === 'undefined'
                || (prefix[2] === 'p' && params.event_value === '1')
                || (prefix[2] === 'r' && params.event_value === '0')) {
              data.push({
                title: field.substring(field.indexOf(':') + 1),
                value: fields[field],
                short: prefix[1] === 'short'
              });
            }
          }
          catch (error) {
            Zabbix.log(4, '[ RocketChat Webhook ] Can\'t parse field "' + field +
              '". The field ignored.');
```

});

```
        }
      });
    }
  }

  return data;
}
```

field_1_full:Host	{HOST.NAME} {{HOST.IP}}	<a href="#">Remove</a>
field_2_short:Severity	{EVENT.SEVERITY}	<a href="#">Remove</a>
field_3_short:Event time	{EVENT.DATE} {EVENT.TIME}	<a href="#">Remove</a>
field_3_short_r:Recovery time	{EVENT.RECOVERY.DATE} {EVE	<a href="#">Remove</a>
field_4_short_r:Event duration	{EVENT.DURATION}	<a href="#">Remove</a>
field_5_short:Operational data	{EVENT.OPDATA}	<a href="#">Remove</a>
field_999_full_p:Trigger descriptio	{TRIGGER.DESCRPTION}	<a href="#">Remove</a>

# ОБЗОР ГЕНЕРАТОРА ПОЛЕЙ

```
addFields: function (fields) {
  var data = [];

  if (typeof fields === 'object' && Object.keys(fields).length) {
    Object.keys(fields)
      .forEach(function(field) {
        if (fields[field] === '') {
          Zabbix.log(4, '[ RocketChat Webhook ] Field "' + field +
            '" can\'t be empty. The field ignored.');
```

```
        }
        else {
          try {
            var parts = field.split(':'),
                prefix = parts[0].split('_');

            if (typeof prefix[2] === 'undefined'
              || (prefix[2] === 'p' && params.event_value === '1')
              || (prefix[2] === 'r' && params.event_value === '0')) {
              data.push({
                title: field.substring(field.indexOf(':') + 1),
                value: fields[field],
                short: prefix[1] === 'short'
              });
            }
          }
          catch (error) {
            Zabbix.log(4, '[ RocketChat Webhook ] Can\'t parse field "' + field +
              '". The field ignored.');
```

```
        }
      });
    }
  }

  return data;
}
```

field_1_full:Host	{HOST.NAME} {HOST.IP}	<a href="#">Remove</a>
field_2_short:Severity	{EVENT.SEVERITY}	<a href="#">Remove</a>
field_3_short:Event time	{EVENT.DATE} {EVENT.TIME}	<a href="#">Remove</a>
field_3_short_r:Recovery time	{EVENT.RECOVERY.DATE} {EVE	<a href="#">Remove</a>
field_4_short_r:Event duration	{EVENT.DURATION}	<a href="#">Remove</a>
field_5_short:Operational data	{EVENT.OPDATA}	<a href="#">Remove</a>
field_999_full_p:Trigger descriptio	{TRIGGER.DESCRPTION}	<a href="#">Remove</a>



# ОБЗОР ГЕНЕРАТОРА ПОЛЕЙ

```
addFields: function (fields) {
  var data = [];

  if (typeof fields === 'object' && Object.keys(fields).length) {
    Object.keys(fields)
      .forEach(function(field) {
        if (fields[field] === '') {
          Zabbix.log(4, '[ RocketChat Webhook ] Field "' + field +
            '" can\'t be empty. The field ignored.');
```

```
        }
        else {
          try {
            var parts = field.split(':'),
                prefix = parts[0].split('_');
```

```
            if (typeof prefix[2] === 'undefined'
              || (prefix[2] === 'p' && params.event_value === '1')
              || (prefix[2] === 'r' && params.event_value === '0')) {
              data.push({
                title: field.substring(field.indexOf(':') + 1),
                value: fields[field],
                short: prefix[1] === 'short'
              });
            }
          }
          catch (error) {
            Zabbix.log(4, '[ RocketChat Webhook ] Can\'t parse field "' + field +
              '". The field ignored.');
```

```
          }
        }
      });
  }

  return data;
}
```

field_1_full:Host	{HOST.NAME} {{HOST.IP}}	<a href="#">Remove</a>
field_2_short:Severity	{EVENT.SEVERITY}	<a href="#">Remove</a>
field_3_short:Event time	{EVENT.DATE} {EVENT.TIME}	<a href="#">Remove</a>
field_3_short_r:Recovery time	{EVENT.RECOVERY.DATE} {EVE	<a href="#">Remove</a>
field_4_short_r:Event duration	{EVENT.DURATION}	<a href="#">Remove</a>
field_5_short:Operational data	{EVENT.OPDATA}	<a href="#">Remove</a>
field_999_full_p:Trigger descriptio	{TRIGGER.DESCRPTION}	<a href="#">Remove</a>

# ФОРМАТ ИМЕНИ ПОЛЕЙ

field_1_full:Host	{HOST.NAME} [{HOST.IP}]	<a href="#">Remove</a>
field_2_short:Severity	{EVENT.SEVERITY}	<a href="#">Remove</a>
field_3_short:Event time	{EVENT.DATE} {EVENT.TIME}	<a href="#">Remove</a>
field_3_short_r:Recovery time	{EVENT.RECOVERY.DATE} {EVE	<a href="#">Remove</a>
field_4_short_r:Event duration	{EVENT.DURATION}	<a href="#">Remove</a>
field_5_short:Operational data	{EVENT.OPDATA}	<a href="#">Remove</a>
field_999_full_p:Trigger descriptio	{TRIGGER.DESCRPTION}	<a href="#">Remove</a>

## field\_1\_short\_p:Host

- **field** - префикс поля для генератора.
- **1** - ID поля. Используется для правильного порядка полей в карточке.
- **short** - определяет должно ли поле быть коротким.
- **p** - опциональный префикс. Определяет, должно ли поле использоваться только для событий проблемы или восстановления. Возможные значения:
  - **p** - problem (проблема)
  - **r** - recovery (восстановление)
- **Host** - имя поля. Будет передано в заголовке поля в карточке, может содержать символы и пробелы.

# ОБЗОР ГЕНЕРАТОРА ПОЛЕЙ

```
addFields: function (fields) {
  var data = [];

  if (typeof fields === 'object' && Object.keys(fields).length) {
    Object.keys(fields)
      .forEach(function(field) {
        if (fields[field] === '') {
          Zabbix.log(4, '[ RocketChat Webhook ] Field "' + field +
            '" can\'t be empty. The field ignored.');
```

```
        }
        else {
          try {
            var parts = field.split(':'),
                prefix = parts[0].split('_');

            if (typeof prefix[2] === 'undefined'
                || (prefix[2] === 'p' && params.event_value === '1')
                || (prefix[2] === 'r' && params.event_value === '0')) {
```

```
          data.push({
            title: field.substring(field.indexOf(':') + 1),
            value: fields[field],
            short: prefix[1] === 'short'
          });
        }
      }
    }
  } catch (error) {
    Zabbix.log(4, '[ RocketChat Webhook ] Can\'t parse field "' + field +
      '". The field ignored.');
```

field_1_full:Host	{HOST.NAME} {{HOST.IP}}	<a href="#">Remove</a>
field_2_short:Severity	{EVENT.SEVERITY}	<a href="#">Remove</a>
field_3_short:Event time	{EVENT.DATE} {EVENT.TIME}	<a href="#">Remove</a>
field_3_short_r:Recovery time	{EVENT.RECOVERY.DATE} {EVE	<a href="#">Remove</a>
field_4_short_r:Event duration	{EVENT.DURATION}	<a href="#">Remove</a>
field_5_short:Operational data	{EVENT.OPDATA}	<a href="#">Remove</a>
field_999_full_p:Trigger descriptio	{TRIGGER.DESCRPTION}	<a href="#">Remove</a>



MEETUP ONLINE '21

# ВОПРОСЫ?

**NAME SURNAME**  
TECHNICAL SUPPORT ENGINEER

**ZABBIX**



MEETUP ONLINE '21

**СПАСИБО!**

**NAME SURNAME**  
TECHNICAL SUPPORT ENGINEER

**ZABBIX**